

Proyecto de Fin de Carrera

**“Proceso de Optimización de bancos de ensayo de  
aviónica”**



**Ingeniería de Telecomunicaciones**

*Universidad Carlos III de Madrid*

*Tutores:*

*Bartolomé Lozano Cerón (EADS-CASA)*

*Harold Molina-Bulla (UC3M)*

*Autor:*

*Álvaro Hernández Benedicto*

*27/Marzo/2009*

## Índice

<b>Capítulo 1.</b>	<b>Introducción</b>	<b>13</b>
1.1.	Objetivo	13
1.2.	¿Qué es un banco de ensayo?	13
1.3.	Motivación	14
1.4.	Ruta de diseño de sistemas de avión	15
1.4.1.	Design (Aircraft Level)	16
1.4.2.	Early Prototyping (Aircraft Level)	17
1.4.3.	Design (System Level)	17
1.4.4.	Virtual Integration (Aircraft Level)	18
1.4.5.	Equipment Level	18
1.4.6.	Integration (System Level)	19
1.4.7.	Physical Integration (Aircraft Level)	21
1.5.	Entorno de proyecto	21
<b>Capítulo 2.</b>	<b>Definición Banco de Ensayo</b>	<b>22</b>
2.1.	Introducción	22
2.1.1.	Proceso de realización de las pruebas	23
2.2.	Subsistema de Banco de Ensayo	24
2.2.1.	Definición del subsistema de ensayo	24
2.2.2.	Funciones de ensayo	27
2.2.2.1.	Graphical Data Visualization System (GDVS)	28
2.2.2.2.	Recording	28
2.2.2.3.	Sequencer Executer	29
2.2.2.4.	Análisis	30
2.2.3.	Simulaciones de SW específico	31
2.2.3.1.	Binding Package	32
2.2.4.	Equipos necesarios y recursos	32
2.2.5.	Tecnologías empleadas en el proceso de optimización	33
2.2.5.1.	CVS	34
2.2.5.2.	Subversion	35
2.2.5.3.	Apache 2.x	36
2.3.	Módulo de Comunicaciones	36
2.3.1.	Chasis	37
2.3.1.1.	VME	37
2.3.1.2.	PXI	40
2.3.2.	Buses de Comunicaciones en aeronáutica	42

2.3.2.1.	ARINC429	43
2.3.2.2.	AFDX	45
2.3.2.3.	MILSTD 1553	48
2.3.2.4.	STANAG 3910	49
2.3.2.5.	CAN	50
2.3.2.6.	DIO	51
2.3.2.7.	AIO	52
2.3.3.	Tarjetas de Comunicaciones	52
2.3.4.	Sensores y Actuadores	54
2.3.5.	Tecnologías empleadas en el proceso de optimización	55
2.3.5.1.	Conmutación	55
2.3.5.2.	Matriz de Conmutación	56
2.3.5.3.	Evolución de comunicaciones en instrumentación	57
<b>2.4.</b>	<b>Módulo de Alimentación</b>	<b>65</b>
2.4.1.	Suministro de Potencia AC	65
2.4.2.	Distribución de Potencia	67
2.4.3.	Suministro de Potencia DC	68
2.4.4.	Suministro de Potencia Ininterrumpida	69
2.4.5.	Control Inteligente de Potencia	69
2.4.6.	Tecnologías empleadas en el proceso de optimización	70
2.4.6.1.	Herramientas de comunicaciones con equipos de control de potencia	70
2.4.6.2.	Software UPS Merge	72
2.4.6.3.	RSH	74
2.4.6.4.	Shutdown Script	74
<b>Capítulo 3. Particularización e Instanciación para una arquitectura genérica de banco de ensayo</b>		<b>75</b>
<b>3.1.</b>	<b>Alcance</b>	<b>75</b>
3.1.1.	Introducción	75
3.1.2.	Evolución de la descripción	75
<b>3.2.</b>	<b>Requerimientos del Banco</b>	<b>76</b>
3.2.1.	Requerimientos HW del Banco	76
3.2.1.1.	Descripción del entorno del banco	77
3.2.1.2.	Diseño – Desarrollo HW	77
3.2.1.3.	Interfaces Humano-Máquina	77
3.2.1.4.	Necesidades de Sincronización	78
3.2.1.5.	Equipos Reales	78
3.2.2.	Requerimientos SW del Banco	79
3.2.2.1.	Equipos Simulados	79

3.2.2.2.	Señales necesarias en el Banco	80
3.2.2.3.	Señales monitorizadas	80
3.2.2.4.	Simulaciones	81
<b>3.3.</b>	<b>Capacidades del Banco</b>	<b>81</b>
3.3.1.	Arquitectura Funcional del Banco	81
3.3.2.	Diagrama de bloques del Banco	81
3.3.3.	Escenarios operativos del Banco	83
3.3.3.1.	Escenario 1	83
3.3.3.2.	Escenario 2	83
3.3.3.3.	Capacidad de conexión del Banco	84
<b>3.4.</b>	<b>Diseño Preliminar del Banco</b>	<b>84</b>
3.4.1.	Descripción física del Banco	84
3.4.2.	Descripción del Sistema de Estimulación y Adquisición	85
3.4.2.1.	Arquitectura SW SEAS	85
3.4.2.2.	Configuración HW SEAS	87
3.4.2.3.	Definición física de entorno SEAS	87
3.4.3.	Base de datos de señales del Sistema	90
3.4.4.	Simulaciones	90
<b>3.5.</b>	<b>Implementación del Banco</b>	<b>90</b>
3.5.1.	Diseño HW detallado	90
3.5.2.	Electrical Testing	92
3.5.3.	Integración SEAS	93
3.5.4.	Configuración SW	93
<b>Capítulo 4.</b>	<b><i>Detección de mejoras y áreas de Optimización</i></b>	<b>95</b>
<b>4.1.</b>	<b>Introducción</b>	<b>95</b>
<b>4.2.</b>	<b>Subsistema de Ensayo</b>	<b>95</b>
<b>4.3.</b>	<b>Modulo de Comunicaciones</b>	<b>96</b>
<b>4.4.</b>	<b>Modulo de Alimentación</b>	<b>98</b>
<b>Capítulo 5.</b>	<b><i>Optimización del Banco de Integración</i></b>	<b>101</b>
<b>5.1.</b>	<b>Introducción</b>	<b>101</b>
<b>5.2.</b>	<b>Subsistema de ensayo: Reorganización de la estructura de ficheros de configuración</b>	<b>101</b>
5.2.1.	Objetivo	101
5.2.2.	Diseño	101
5.2.2.1.	Necesidades de subsistema de ensayo	101

5.2.2.2.	Elección del sistema de gestión de versiones	102
5.2.2.3.	Integración	103
5.2.3.	Resumen de mejoras	106
<b>5.3.</b>	<b>Módulo de Comunicaciones: Inclusión de Matrices de Conmutación y Equipos de Análisis</b>	<b>107</b>
5.3.1.	Objetivo	107
5.3.2.	Rediseño de la estructura de banco	108
5.3.2.1.	Elección de la tecnología	108
5.3.2.2.	Elección de Matrices	109
5.3.3.	Diseño sobre un caso particular	113
5.3.3.1.	Necesidades de conmutación	114
5.3.3.2.	Necesidades de configuración	115
5.3.3.3.	Diseño particular	116
5.3.4.	Resumen de mejoras	119
<b>5.4.</b>	<b>Módulo de Alimentación: Apagado Remoto y Ordenado de Bancos de Ensayo</b>	<b>122</b>
5.4.1.	Objetivo	122
5.4.2.	Arquitecturas de Banco	123
5.4.2.1.	Banco con SAI y Remote Power Control	123
5.4.2.2.	Banco con SAI	124
5.4.2.3.	Banco con Remote Power Control	125
5.4.2.4.	Banco sin recursos de apagado adicionales	126
5.4.3.	Apagado ordenado desde SAI	126
5.4.3.1.	SAI	126
5.4.3.2.	Servidor	128
5.4.3.3.	Script	130
5.4.3.4.	Configuración	130
5.4.3.5.	Ejemplo de Shutdown	133
5.4.3.6.	Corte de alimentación	134
5.4.4.	Resumen de mejoras	135
<b>5.5.</b>	<b>Funcionalidad conjunta de las mejoras propuestas</b>	<b>135</b>
<b>Capítulo 6.</b>	<b>Conclusiones</b>	<b>137</b>
<b>6.1.</b>	<b>Trabajos Futuros</b>	<b>138</b>
<b>Anexo I.</b>	<b>Descripción de las herramientas del subsistema de ensayo</b>	<b>140</b>
<b>AI.i.</b>	<b>Introducción</b>	<b>140</b>
<b>AI.ii.</b>	<b>Graphical Data Visualization System</b>	<b>140</b>

AI.iii.	Recording	143
AI.iv.	Análisis	145
<b>Anexo II.</b>	<b>Descripción de las herramientas de simulación SEAS</b>	<b>147</b>
AII.i.	Introducción	147
AII.ii.	Simulaciones ADA	147
AII.ii.a.	Configuración	147
AII.ii.b.	Compilación	149
AII.ii.c.	Ejecución de la simulación	151
AII.ii.d.	Debugging	153
AII.iii.	Simulaciones Non Ada	153
AII.iii.a.	Compilación	159
AII.iii.b.	Ejecución de la simulación	159
<b>Anexo III.</b>	<b>Descripción de buses de aviónica</b>	<b>160</b>
AIII.i.	Introducción	160
AIII.ii.	Arinc 429	160
AIII.iii.	AFDX	166
AIII.iv.	MIL-STD-1553	172
AIII.v.	Stanag 3910	178
AIII.vi.	CAN	180
<b>Anexo IV.</b>	<b>Tarjetas de comunicaciones empleadas</b>	<b>187</b>
AIV.i.	AMC 1553	187
AIV.ii.	AMC 429	189
AIV.iii.	AVI 3910	191
AIV.iv.	PCI-CAN/XS2	193
<b>Anexo V.</b>	<b>Redes de conexión en matrices de conmutación</b>	<b>195</b>
AV.i.	Introducción	195
AV.ii.	Clasificación por puntos de cruce	195
AV.iii.	Infrautilización	197
AV.iv.	Clasificación por tecnología de conmutación	197
<b>Anexo VI.</b>	<b>Proceso de desarrollo de bancos de integración</b>	<b>201</b>

AVI.i.	Introducción	201
AVI.ii.	Project Management	201
AVI.iii.	Análisis de requerimientos	204
AVI.iv.	Diseño preliminar	205
AVI.v.	Implementación del Banco	208
AVI.vi.	Consecución y Subcontratación	211
AVI.vii.	Verificación	213
AVI.viii.	Validación	215
AVI.ix.	Proceso de configuración	217
AVI.x.	Documentación	219
AVI.xi.	Soporte	219
<b>Anexo VII.</b>	<b>Resultado de pruebas del banco</b>	<b>222</b>
AVII.i.	Introducción	222
AVII.ii.	Resultados pruebas de continuidad	222
AVII.iii.	Resultados pruebas de funcionalidad	259
<b>Anexo VIII.</b>	<b>Planos de los bancos de integración.</b>	<b>273</b>
AVIII.i.	Introducción	273
AVIII.ii.	Descripción del banco	273
<b>Anexo IX.</b>	<b>Teoría sobre el calentamiento global</b>	<b>276</b>
AIX.i.	Introducción	276
AIX.ii.	Fenómeno Natural	276
AIX.iii.	Organización Internacional	277
AIX.iii.a.	Protocolo de Kyoto	277
<b>Anexo X.</b>	<b>Código de interacción con BlackBox</b>	<b>280</b>
AX.i.	Introducción	280
AX.ii.	Código	280
<b>Anexo XI.</b>	<b>Pruebas de comprobación del banco</b>	<b>284</b>
AXI.i.	Electrical Testing	284

AXI.i.a. Prueba de continuidad	284
AXI.i.b. Prueba de continuidad cruzada	286
AXI.i.c. Prueba de aislamiento respecto a carcasa	287
<b>AXI.ii. Integración de SEAS</b>	<b>289</b>
AXI.ii.a. Prueba de inspección visual	289
AXI.ii.b. Prueba funcional	290
<b>Anexo XII. Código de integración de SVN</b>	<b>292</b>
AXII.i. Update	292
AXII.ii. Commit	293
<b>Anexo XIII. Script de apagado remoto</b>	<b>295</b>
AXIII.i. Banco con SAI y RPC	295
AXIII.ii. Banco con RPC	295
AXIII.iii. Banco sin recursos adicionales	297
<b>Capítulo 7. Bibliografía</b>	<b>299</b>



## Índice de Figuras

<i>Figura 1 Diseño en V de producto [17]</i>	16
<i>Figura 2 Esquema de Pruebas [18]</i>	22
<i>Figura 4 Proceso de realización de Test [18]</i>	24
<i>Figura 5 Esquema de test con Subsistema de Ensayo particular [18]</i>	25
<i>Figura 6 Representación de recursos [18]</i>	33
<i>Figura 7 Características VME</i>	40
<i>Figura 8 Buses de sincronización y disparo [26]</i>	41
<i>Figura 9 Integración PXI y cPCI [27]</i>	42
<i>Figura 10 Tipos de equipos ARINC429 [28]</i>	44
<i>Figura 11 Esquema Switched Ethernet AFDX [31]</i>	47
<i>Figura 12 Configuración AFDX [30]</i>	47
<i>Figura 13 Esquema MIL1553 [33]</i>	49
<i>Figura 14 Bus Concept 3910 [36]</i>	50
<i>Figura 15 Topología CAN [37]</i>	51
<i>Figura 16 Conector GPIB [43] [44]</i>	58
<i>Figura 17 Conector RS-232 [45], [46]</i>	59
<i>Figura 18 Conector RJ45 [47], [48]</i>	60
<i>Figura 19 Conector USB Tipo A [49], [50]</i>	61
<i>Figura 20 Arquitectura LXI</i>	64
<i>Figura 21 Latencia vs. Ancho de Banda [26]</i>	65
<i>Figura 22 Frontal de alimentación</i>	66
<i>Figura 23 Power Distribution</i>	68
<i>Figura 24 Fuente 28VDC</i>	68
<i>Figura 25 SAI MGE</i>	69
<i>Figura 26 Pulizzi IPC3401-NET</i>	70
<i>Figura 27 BlackBox Web Configuration</i>	72
<i>Figura 28 Arquitectura de apagado [51]</i>	73
<i>Figura 29 Arquitectura del Banco</i>	82
<i>Figura 30 Arquitectura SEAS [19]</i>	85
<i>Figura 31 Diagrama de Arquitectura HW [19]</i>	89
<i>Figura 32 Diagrama de bloques del banco</i>	91
<i>Figura 32 Banco Frontal</i>	92
<i>Figura 35 Estructura de ficheros</i>	104
<i>Figura 36 Puntos de Cruce vs. Precio</i>	113
<i>Figura 37 Configuración de matrices</i>	114
<i>Figura 38 Conexiones de la matriz</i>	116
<i>Figura 39 Diseño con matriz</i>	117
<i>Figura 40 Diseño con matriz y sistemas reales</i>	118

<i>Figura 41</i>	<i>Diseño con matriz y subsistema simulado</i>	<i>119</i>
<i>Figura 42</i>	<i>Diseño con nuevo sistema</i>	<i>120</i>
<i>Figura 43</i>	<i>Integración de múltiples subsistemas</i>	<i>121</i>
<i>Figura 44</i>	<i>Arquitectura SAI y Remote Power Control</i>	<i>124</i>
<i>Figura 45</i>	<i>Arquitectura SAI</i>	<i>125</i>
<i>Figura 46</i>	<i>Arquitectura Remote Power Control</i>	<i>125</i>
<i>Figura 47</i>	<i>Arquitectura sin HW adicional</i>	<i>126</i>
<i>Figura 48</i>	<i>Circuitos de ROO y RPO</i>	<i>127</i>
<i>Figura 49</i>	<i>Lógica de marcha/parada de SAI's auxiliares.</i>	<i>128</i>
<i>Figura 50</i>	<i>NMP Web Page</i>	<i>130</i>
<i>Figura 51</i>	<i>NMP Shutdown Configuration</i>	<i>132</i>
<i>Figura 52</i>	<i>MGE SAI SYSTEMS - Settings</i>	<i>133</i>
<i>Figura 53</i>	<i>Ejemplo de shutdown [51]</i>	<i>134</i>
<i>Figura 54</i>	<i>Composición de Paneles GVDS</i>	<i>141</i>
<i>Figura 55</i>	<i>Proceso INIT de Simulación</i>	<i>150</i>
<i>Figura 56</i>	<i>Proceso EXEC de Simulación</i>	<i>150</i>
<i>Figura 57</i>	<i>Proceso STOP de Simulación</i>	<i>151</i>
<i>Figura 58</i>	<i>Pantalla de control de simulación por GVDT</i>	<i>152</i>
<i>Figura 59</i>	<i>Pantalla de control de simulación por Aplicación</i>	<i>153</i>
<i>Figura 60</i>	<i>Características ARINC429</i>	<i>161</i>
<i>Figura 61</i>	<i>Modulación ARINC429</i>	<i>161</i>
<i>Figura 62</i>	<i>Características eléctricas ARINC429</i>	<i>162</i>
<i>Figura 63</i>	<i>Configuración 1 ARINC429</i>	<i>163</i>
<i>Figura 64</i>	<i>Configuración 2 ARINC429</i>	<i>163</i>
<i>Figura 65</i>	<i>Ejemplo trama BCD ARINC429</i>	<i>164</i>
<i>Figura 66</i>	<i>Ejemplo trama BNR ARINC429</i>	<i>164</i>
<i>Figura 67</i>	<i>Discrete Bits ARINC429</i>	<i>165</i>
<i>Figura 68</i>	<i>Matriz de Status ARINC429</i>	<i>165</i>
<i>Figura 69</i>	<i>Redundancia AFDX</i>	<i>167</i>
<i>Figura 70</i>	<i>Virtual Link AFDX</i>	<i>168</i>
<i>Figura 71</i>	<i>Virtual Link AFDX 2</i>	<i>168</i>
<i>Figura 72</i>	<i>BAG</i>	<i>169</i>
<i>Figura 73</i>	<i>FSS AFDX</i>	<i>171</i>
<i>Figura 74</i>	<i>Data Set AFDX</i>	<i>171</i>
<i>Figura 75</i>	<i>Intercalado de tramas AFDX</i>	<i>172</i>
<i>Figura 76</i>	<i>Trama Ethernet</i>	<i>172</i>
<i>Figura 77</i>	<i>Command Word 1553</i>	<i>173</i>
<i>Figura 78</i>	<i>Status Word 1553</i>	<i>173</i>
<i>Figura 79</i>	<i>Data Word 1553</i>	<i>174</i>
<i>Figura 80</i>	<i>Codificación Manchester 1553</i>	<i>174</i>

<i>Figura 81 Timing 1553</i>	175
<i>Figura 82 Envío BC-RT 1553</i>	175
<i>Figura 83 Envío RT-BC 1553</i>	176
<i>Figura 84 Envío RT-RT 1553</i>	176
<i>Figura 85 Modo comando sin datos 1553</i>	176
<i>Figura 86 Modo comando con datos 1553</i>	177
<i>Figura 87 Modo comando con datos rx 1553</i>	177
<i>Figura 88 Otros modos 1553</i>	177
<i>Figura 89 HS Action Word 3910</i>	178
<i>Figura 90 HS Status Word 3910</i>	179
<i>Figura 91 HS Message Word 3910</i>	179
<i>Figura 92 Bit-Wise Arbitration CAN</i>	180
<i>Figura 93 Data Frame CAN</i>	181
<i>Figura 94 Data Frame 1 CAN</i>	182
<i>Figura 95 Data Frame 2 CAN</i>	182
<i>Figura 96 Remote Frame 3 CAN</i>	183
<i>Figura 97 Error Frame CAN</i>	184
<i>Figura 98 Overload Frame CAN</i>	184
<i>Figura 99 Características CAN</i>	185
<i>Figura 100 Capa Física CAN</i>	186
<i>Figura 101 AMC1553 [39]</i>	187
<i>Figura 102 Esquema AMC1553 [39]</i>	189
<i>Figura 103 Esquema API429 [40]</i>	191
<i>Figura 104 Esquema AVI3910 [41]</i>	193
<i>Figura 105 PCI/CAN XS2 [42]</i>	194
<i>Figura 106 Puntos de cruce NxM [1]</i>	195
<i>Figura 107 Puntos de cruce de N [1]</i>	196
<i>Figura 108 Puntos de N plegada [1]</i>	196
<i>Figura 109 Esquema de SSR [56]</i>	198
<i>Figura 110 Conmutador CMOS</i>	199
<i>Figura 111 Relé Reed [57]</i>	200
<i>Figura 112 Proceso de construcción de banco [58]</i>	201
<i>Figura 113 Subrack 1</i>	274
<i>Figura 114 Subrack 2</i>	275
<i>Figura 115 Relación de acumulación de CO2 y aumento de temperatura [54]</i>	279

## **Índice de Tablas**

<i>Tabla 2 Escenario 1</i>	83
<i>Tabla 3 Escenario 2</i>	84
<i>Tabla 4 Descripción de partes</i>	88
<i>Tabla 5 Distribución de tarjetas</i>	89
<i>Tabla 6 Estudio del coste de desarrollo de bancos</i>	98
<i>Tabla 7 Estudio de consumo de potencia de banco</i>	99
<i>Tabla 8 Estudio de mejora en la contaminación</i>	99
<i>Tabla 9 Comparativa matrices de conmutación en el mercado</i>	112
<i>Tabla 10 Resultados M-A97</i>	223
<i>Tabla 11 Resultados M-A98</i>	224
<i>Tabla 12 Resultados M-A99</i>	225
<i>Tabla 13 Resultados M-A100</i>	226
<i>Tabla 14 Resultados M-A101</i>	227
<i>Tabla 15 Resultados M-A08</i>	229
<i>Tabla 16 Resultados M-A10</i>	232
<i>Tabla 17 Resultados M-A14</i>	234
<i>Tabla 18 Resultados M-A15</i>	238
<i>Tabla 19 Resultados M-A16</i>	239
<i>Tabla 20 Resultados M-A17</i>	244
<i>Tabla 21 Resultados M-A04</i>	248
<i>Tabla 22 Resultados M-A07</i>	255
<i>Tabla 23 Resultados M-A22</i>	256
<i>Tabla 24 Resultados M-A23</i>	257
<i>Tabla 25 Resultados M-A24</i>	257
<i>Tabla 26 Resultados M-A96</i>	258
<i>Tabla 27 Resultados visuales "Subrack 1"</i>	259
<i>Tabla 28 Resultados funcionales "Subrack 1"</i>	272
<i>Tabla 29 Prueba de continuidad</i>	285
<i>Tabla 30 Prueba de continuidad cruzada</i>	287
<i>Tabla 31 Prueba de aislamiento respecto a carcasa</i>	289
<i>Tabla 32 Prueba de inspección visual</i>	290
<i>Tabla 33 Prueba funcional</i>	291

## Capítulo 1. Introducción

### 1.1. Objetivo

El objetivo de este proyecto, “**Proceso de optimización de bancos de ensayo de aviónica**”, es descubrir y solucionar los problemas más importantes detectados en un banco de pruebas, para generalizar estas soluciones e introducirlas en los futuros proyectos de bancos de ensayo.

La detección y optimización del banco de ensayo exige una división clara de la estructura del proyecto en cuatro partes:

1. **Definición de un banco de ensayo genérico.** Análisis de la estructura genérica y las tecnologías habitualmente empleadas en el banco de pruebas con el objetivo de conocer la composición del banco.
2. **Particularización e instanciación para una arquitectura genérica de banco de ensayo.** Definición de una estructura más concreta sobre lo que debe ser y cómo de se debe definir un banco de pruebas.
3. **Detección de áreas de optimización.** Análisis de los problemas encontrados en las diferentes partes de las que consta el banco.
4. **Optimización de las áreas analizadas.** Solución de los problemas detectados en el punto anterior, generalizando lo más posible las soluciones.

### 1.2. ¿Qué es un banco de ensayo?

Un **banco de ensayo** es una estructura real o simulada de una pequeña parte de avión creado para realizar pruebas en tierra de esta parte. Las pequeñas partes en que se divide el avión para realizar las pruebas, se denominan **subsistemas**.

La división en subsistemas se realiza de modo que las pruebas sean progresivas:

1. Pruebas sobre los subsistemas divididos.
2. Integración de subsistemas en sistemas mayores.
3. Integración de sistemas en conjuntos más completos.

Un banco de ensayo puede ser **SoftWare** (SW), **HardWare** (HW) o HW/SW.

- **Banco de ensayo software:** Está basado en la simulación software de todos los elementos de un subsistema de avión. El objetivo es realizar pruebas sobre la estructura teórica de avión.
- **Banco de ensayo hardware:** Está basado en la implementación de la estructura de un subsistema del avión incluyendo tantas partes reales de éste como se pueda. Las partes necesarias para realizar las pruebas del subsistema que no tiene sentido tener físicamente en tierra (no existe un movimiento real de avión) serán simuladas por software.
- **Banco de ensayo HW/SW:** Está basado en la implementación de un subsistema de avión. Un subsistema está a su vez compuesto de muchas partes, aquí se subdivide éste aún más para realizar las pruebas sobre una parte concreta de él, y simulando todo el entorno que necesita, pero que en ese momento no interesa tener físicamente.

### 1.3. Motivación

Debido a la gran variedad de sistemas de que dispone un avión, y la gran cantidad de partes que deben ser verificadas y validadas, los bancos de ensayo son diversos, y tanto la creación, como el ensamblado y el uso, pueden generar soluciones no óptimas de bancos de ensayo.

Actualmente se está trabajando en la generalización y apertura de los bancos de ensayo, para conseguir el mejor uso de todos los recursos de la empresa, y facilitar todo lo posible al usuario final de los bancos, la realización de las pruebas sobre él.

El mundo de los bancos de ensayo para pruebas es muy amplio en el entorno de la ingeniería, no sólo para cumplir todas las condiciones de seguridad exigidas por las diferentes instituciones, si no que siempre es necesario probar los diseños teóricos antes de realizar una implementación real de éste. Además, teniendo unos modelos teóricos y bancos de ensayo completamente representativos del equipo final, se podrán aplicar mejoras sobre éstos sabiendo que se está repercutiendo del mismo modo que en el modelo final.

Los bancos de ensayo se emplean tanto en sector automovilístico, como la arquitectura, como en muchos otros sectores de la ingeniería, pero son extremadamente importantes en sectores de unas exigencias de seguridad altísimas, como es el sector de la aviónica.

Gracias a la colaboración de la empresa EADS-CASA se podrá realizar un proyecto del campo de los bancos de ensayo desde dentro, obteniendo toda la información necesaria y permitida tanto para analizar la estructura de éstos, como para realizar la optimización del diseño de los bancos.

#### **1.4. Ruta de diseño de sistemas de avión**

Para ver el marco en el que se sitúa el proyecto es preciso seguir la línea de producción de un avión. Así se puede conocer cuál es el proceso de creación del avión y situarse en la capa de los bancos de ensayo para conocer más sobre cuál es la integración de los bancos de ensayo dentro de la cadena de creación del avión.

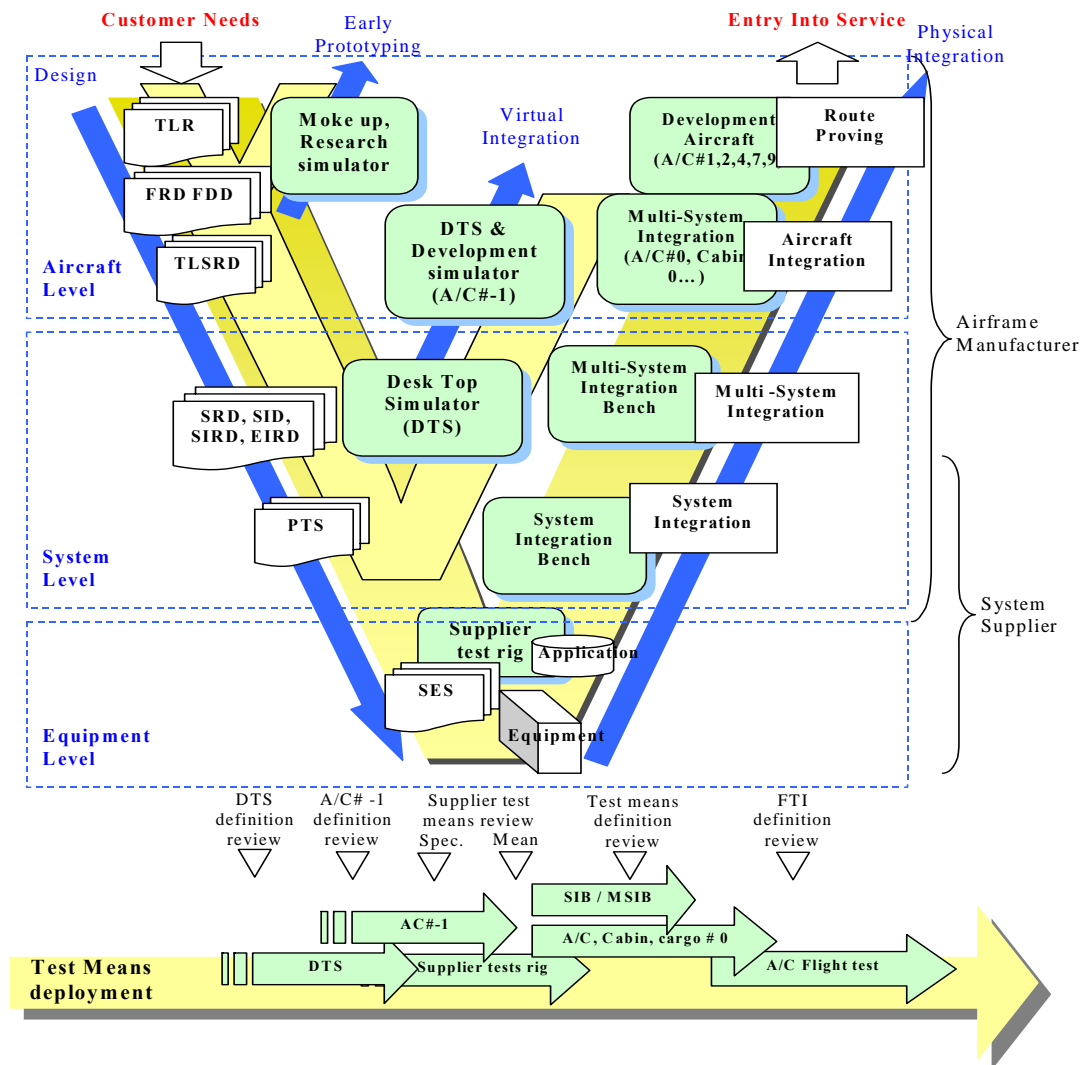


Figura 1 Diseño en V de producto [17]

Como se puede ver en el gráfico, hay 3 niveles que seguir en la línea de diseño. Es necesario seguir esta línea de diseño para garantizar las especificaciones de seguridad del producto y las exigencias de cada cliente.

A través de este esquema se definen diferentes proyectos SW, HW y SW/HW que sirven como paso de pruebas de equipos, sistemas y subsistemas de avión.

#### 1.4.1. Design (Aircraft Level)



En este punto se obtiene una primera aproximación de diseño de avión a partir de los requerimientos de cliente. No se obtienen informes ni diseños muy técnicos, se hace la primera interacción entre las necesidades exigidas por el cliente, y los recursos necesarios para cubrir las necesidades.

Las subdivisiones de cada uno de los diferentes sistemas involucrados en el diseño final, son los que se especifican y diseñan minuciosamente para se comprobados en el *system level*.

#### **1.4.2. Early Prototyping (Aircraft Level)**

El primer prototipo es el resultado de la cobertura de las necesidades especificadas por el cliente con una descripción general sobre los recursos que se verán involucrados. Es un primer paso de realimentación con el cliente, para valorar la línea de actuación.

#### **1.4.3. Design (System Level)**

Tras comprobar que las especificaciones definidas se han cumplido, se realizan los primeros modelos de sistema o *engineering simulation*.

En la primera fase del nivel de sistema se realizan los modelos **DeskTop Simulator (DTS)** de cada sistema del avión. Éstos son modelos SW de los subsistemas de avión y servirán como primera fase de realimentación con el cliente.

En este paso se modelan los sistemas de avión, incluyendo las entradas necesarias para probarlo y entradas adicionales para simular errores, así como las salidas necesarias para comprobar el correcto funcionamiento y unir todos los sistemas de avión con el objetivo de simular en SW el sistema completo. Se estimulan un conjunto de entradas y se comprueba el conjunto de salidas, con lo que probaremos el funcionamiento teórico de los modelos realizados.

Con estos modelos se realizan las pruebas SW para conocer si el funcionamiento previsto en la primera fase se cumple, y detectar los posibles errores existentes. Se realizan modelos de todos los subsistemas, incluyendo si es necesario todas las leyes físicas que dirigen el comportamiento del avión y los equipos.

En un primer momento se realizan modelos de cada subsistema individual para poder comprobar el funcionamiento o en su caso detectar los problemas localmente. Posteriormente se construye la integración de los sistemas SW en un diseño virtual, obteniendo así un primer modelo completo de avión, o modelos completos de sistemas suficientemente independientes en los que realizar las simulaciones y test necesarios para garantizar las especificaciones del cliente y la correcta integración de las diferentes partes.

#### **1.4.4. Virtual Integration (Aircraft Level)**

Este diseño está basado en el desarrollo individual de cada subsistema mediante modelos SW y la múltiple integración de ellos para obtener un modelo sobre el que comprobar con el cliente el adecuado funcionamiento.

Es una integración virtual en modelos SW, por lo que el chequeo de necesidades por parte del cliente se basará en las implementaciones teóricas del diseño de avión, sin estar involucrada ninguna estructura real.

#### **1.4.5. Equipment Level**

En este punto, tras la verificación y validación (V&V)<sup>1</sup> de los modelos teóricos se pasa a la realización de bancos de ensayo para la verificación de los requerimientos de los equipos reales de avión por separado.

---

<sup>1</sup> V&V: Verificación significa: ¿Estamos haciendo bien el producto? (Are we building the product right?)

Validación significa: ¿Estamos haciendo el producto que nos piden? (Are we building the right product?)

Cada equipo de avión deberá cumplir unas especificaciones de potencia, seguridad ante fallos, velocidad, etc., por lo que es necesario comprobar que cada equipo hace realmente lo que especifica.

Estos bancos comprueban el funcionamiento de los equipos, pero un equipo siempre irá montado junto con muchos otros, por lo que es necesario simular todos los equipos de su entorno mediante SW. Los equipos simulados cumplirán con las especificaciones teóricas, por lo que permite conocer el comportamiento del equipo en pruebas estando en un entorno teórico perfecto.

Estos bancos, igual que antes, no son sólo necesarios para esta comprobación previa de los equipos antes de montarlos en avión. Ante un fallo de un equipo en un avión real que ya está en uso, se puede desmontar este equipo del avión, integrarlo en el banco de ensayo y simular las condiciones que originaron su error, para comprobar así que fue lo que falló.

#### **1.4.6. Integration (System Level)**

En esta fase del nivel de sistema, tras comprobar que se cumplen las especificaciones de los equipos de avión, se realiza la integración de sistemas en bancos de ensayo (*HW/SW Integration Bench*) para verificar los requerimientos de equipos de avión.

La mejor forma de probar el sistema completo es realizar un modelo escalar, subdividiendo todo el avión en subsistemas tan concretos como se necesite y probando estos por separado, hasta tener la certeza de que todo funciona de manera aislada, y entonces realizar la integración completa.

El primer objetivo es probar el subsistema, simulando con SW el resto de partes (subsistemas y sistemas que actúen directamente con él y señales necesarias) que formarán el sistema final. De nuevo se pretende realizar testes del subsistema, por lo que se deberá configurar un escenario capaz de verificar

las características esperadas de éste, simulando aquellas partes de las que no se pueda o quiera disponer todavía.

Posteriormente se realiza una integración de los equipos en bancos (***S**ystem *Integration* **B**ench*) para verificar el correcto funcionamiento del conjunto. El objetivo ahora es verificar el funcionamiento tanto de equipos como de los sensores y actuadores sobre el conjunto de equipos de avión. Ya se tiene el subsistema real y se integrará con todas las partes del avión que puedan ser reales, simulando aquello que no podamos disponer.

Este es un paso previo al montaje en avión, por lo que aunque todavía no está en el entorno real de avión, es necesario tener todas las partes reales que sean posibles, y sólo simular en software aquello imposible de montar en tierra. Ahora sí que se integran todos los equipos reales pertenecientes a un sistema, ya que cada uno tendrá sus particularidades. Lo que se asegura, es el correcto funcionamiento del conjunto, incluyendo equipos de alimentación y otros equipos concretos del sistema, así como las situaciones concretas de funcionamiento (temperatura, presión...)

Cuando está comprobado el funcionamiento correcto y completo de cada sistema real, se realiza la integración final en avión para realizar las últimas pruebas sobre el funcionamiento de todo el conjunto de avión. En este punto se asciende en el esquema al nivel de avión.

Los bancos de ensayo no son sólo necesarios para probar el funcionamiento previo de los equipos, sino que también se emplean para comprobar un sistema ya montado en un avión, que en un momento concreto falla. Se desmontan los equipos del avión y se montan en el banco, pudiendo realizar todas las pruebas necesarias sobre ellos en un entorno controlado, más barato y con la capacidad de repetición de las condiciones que sean necesarias.

#### **1.4.7. Physical Integration (Aircraft Level)**

Este es el último paso hacia la aprobación del modelo final de avión. La integración de todos los subsistemas en el avión se realiza de modo escalar, comenzando por los subsistemas más básicos para realizar el vuelo, e incorporando progresivamente todas las necesidades exigidas por el cliente.

En cada integración se realizan las pruebas de ensayos en vuelo. Al igual que en el *system level*, se realizan ensayos en tierra sobre todos los subsistemas, al incluirlos en avión, se realizarán el conjunto de pruebas en vuelo.

Este paso en ningún caso se realiza de forma brusca incluyendo los sistemas simultáneamente en vuelo, si no que se prueban ciertos sistemas en otros aviones para comprobar el correcto funcionamiento. Siempre es necesario realizar pruebas intermedias entre *system level* y *aircraft level*. Todos los subsistemas probados en los bancos de ensayo pasarán por un proceso previo de prueba en tierra que garantice la correcta integración con todo el sistema de aviónica del avión.

#### **1.5. Entorno de proyecto**

El proyecto se enmarca dentro del *system level*.

Es en este punto donde se encuentran la mayor parte de los bancos de ensayo, ya que es aquí dónde se exigen las pruebas sobre los subsistemas individuales de avión.

Las tecnologías empleadas en los bancos de ensayo serán las mismas que finalmente irán en avión, ya que es necesario mantener un sistema fidedigno de lo que realmente se empleará.

## Capítulo 2. Definición Banco de Ensayo

### 2.1. Introducción

Un banco de ensayo es un sistema de pruebas que ofrece el entorno necesario para realizar el proceso de V&V sobre un equipo, subsistema o sistema de A/C a lo largo de toda la cadena de creación.

El concepto que se pretende conseguir con los bancos de ensayo se expresa de forma gráfica en la siguiente figura.

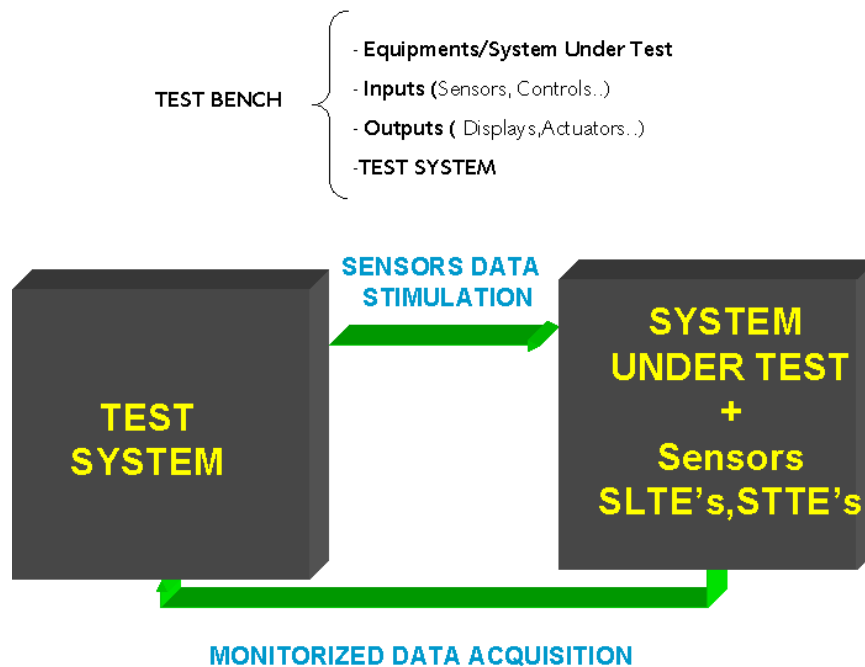


Figura 2 Esquema de Pruebas [18]

En este apartado se va a realizar una definición completa de las partes de que se compone un banco de ensayo, definiendo tanto la estructura como las características más importantes que lo representan.

Para realizar correctamente el proceso de se necesita una correcta integración de las diferentes partes que forman el banco de ensayo.

Se divide el banco en tres partes:

- Subsistema de ensayo.
- Módulo de comunicaciones.
- Módulo de alimentación.

Se analizará cada una de las partes del banco desde el punto de vista de las tecnologías y necesidades empleadas actualmente, y las que se emplearán en las mejoras propuestas.

### **2.1.1. Proceso de realización de las pruebas**

En la creación de un producto de A/C, los campos de pruebas que se requieren son los siguientes [18]:

- Simulaciones de ingeniería (Sistemas de pruebas virtuales)
- Pruebas de entorno
- Verificación de equipos SW
- Bancos de pruebas funcionales
- RIG de pruebas
- Pruebas de equipos ya construidos

Para cada uno de estos entornos, el esquema de realización de pruebas que se requiere es el mismo. Independientemente de que sea una prueba para un equipo o para un sistema, e independientemente del tamaño del banco, los puntos que hay que seguir son los mismos:

- Programación del test de pruebas.
- Definición del test con las señales necesarias (señales del ICD de avión (A/C) y señales de simulación), escoger las simulaciones necesarias y los casos de pruebas a simular.
- Configuración y realización del test con las pruebas definidas.
- Obtener los datos guardados en el test, y realizar el post-procesado para verificar los requerimientos iniciales.

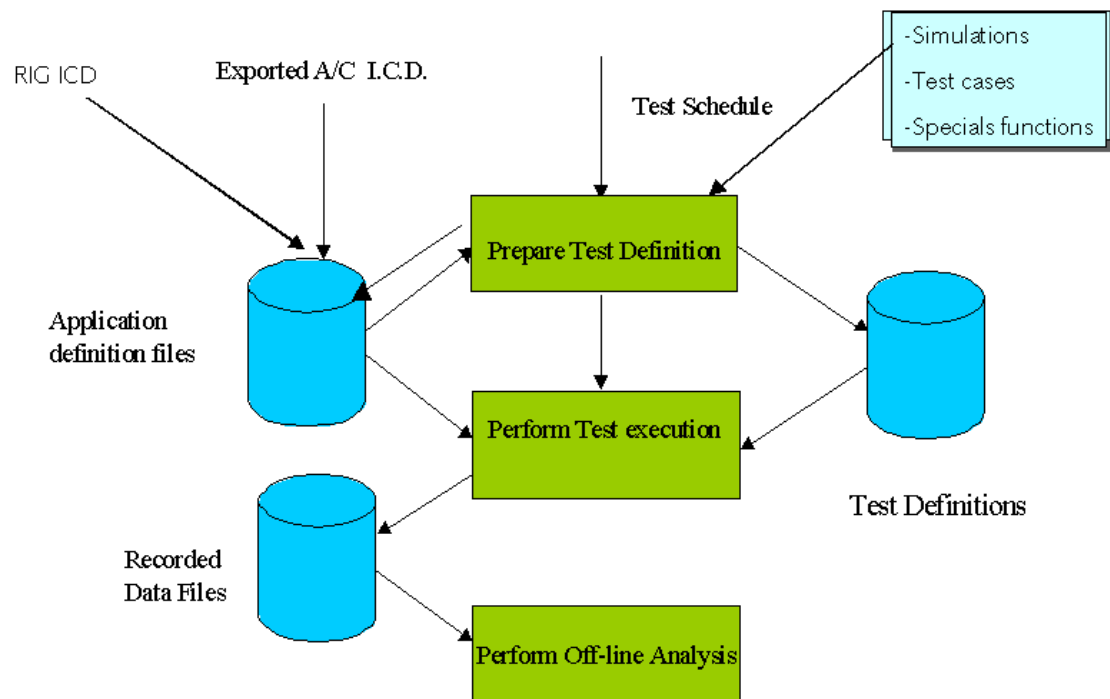


Figura 3 Proceso de realización de Test [18]

## 2.2. Subsistema de Banco de Ensayo

### 2.2.1. Definición del subsistema de ensayo

El subsistema de banco de ensayo es un **Sistema** genérico de Estimulación, **Adquisición** y **Simulación** (SEAS) [18] para conseguir la visualización e interacción con las señales del banco.

SEAS es un sistema en tiempo real de ejecución de test sobre bancos de pruebas de sistemas para SW A/C, desarrollo de sistemas e integración de pruebas, **basadas en tierra**.

Aporta un entorno capaz de soportar pruebas y / o simulaciones de modelos y diseños basados en equipos, subsistemas o sistemas de aviónica, o para equipos reales. SEAS puede ser empleado en cualquier proyecto



definiendo correctamente las señales, simulaciones, topologías de buses y funciones particulares del proyecto.

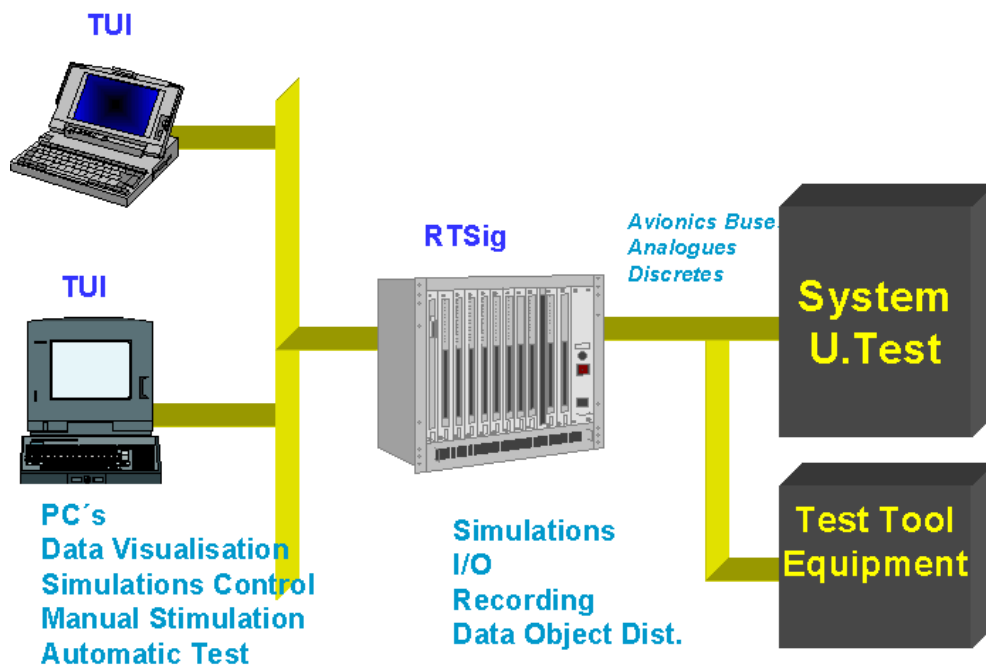


Figura 4 Esquema de test con Subsistema de Ensayo particular [18]

SEAS está orientado entorno al concepto de señal. Las señales son extraídas del ICD de A/C y definidas en un fichero “.xml” con las características:

- Interfaz físico (canal, tarjeta, bus...)
- Escalado (*raw* / value)

SEAS aporta la estructura necesaria para la realización de pruebas en cualquier entorno de simulación:

- Simulación de Ingeniería.
- Banco de SW.
- Banco funcional.
- RIG.

Los fundamentos en los que se basa para ser usado en cualquier banco de ensayo SW o HW/SW son [19]:

- Abierto, distribuido y escalable.
- Entorno de simulación multiplataforma y multi-SO.
- Soporte para cualquier lenguaje de programación.
- Capacidad de múltiples tarjetas VME, PCI, PCI-extended y soporte USB para otros periféricos.
- Arquitectura abierta de pruebas basada en *Switched Area Network* paralela a la arquitectura de A/C.
- Capacidad de integración de:
  - Importación de ICD de A/C
  - Simulaciones & Modelos ya creados
  - *Scripts* de V&V
  - Herramientas de pruebas industriales comerciales
  - Tarjetas de aviónica I/F comerciales
  - Procesadores comerciales
  - Herramientas HMI comerciales
  - Diferentes productos multimedia comerciales
  - Instrumentación comercial
- Seguimiento de las necesidades de V&V.
- Adaptabilidad para diferentes rutas de diseño de proyectos A/C.
- Capacidad de integración de Simulaciones SW con el entorno HW

El subsistema de ensayo constará a su vez de 2 subsistemas:

#### I. Subsistema de estimulación y simulación

Estará implementado en alguna de las tecnologías de chasis soportadas, como VME o PXI. Cada chasis puede tener múltiples tarjetas procesadoras y las tarjetas de Input/Output (I/O) necesarias y podrá aceptar una plataforma multiprocesador.

Estos chasis estarán instalados en una o varias cabinas interconectadas por red o cableados punto a punto. Además las tarjetas de comunicaciones y

procesadoras estarán conectados con los equipos o sistemas bajo prueba mediante los correspondientes interfaces A/C (ARINC429, MILSTD1553, STANAG3910...) o del laboratorio (Bus Serie, USB, red...).

Gracias a la flexibilidad de SEAS, se permite reutilizar simulaciones o partes de simulaciones anteriores para nuevas pruebas, como pueden ser radares simulados, sistemas de navegación, modelos de avión...

## II. Subsistema de usuarios (User Interface Subsystem)

Para utilizar el subsistema de ensayo, los usuarios pueden estar tanto presentes en el laboratorio, como de forma remota (en la red local). Pueden realizar estén donde estén, definiciones, control de test en funcionamiento, análisis de datos..., a través de las múltiples funciones gráficas y de visualización de datos disponibles en SEAS.

No existe la necesidad de estar conectado al banco para realizar todas las tareas, si no que se ofrecen utilidades para trabajar *off-line*:

- Configuración de nodos.
- Definición de señales.
- Definición de test.
- Creación de paneles de visualización.

Además se ofrecen todas las capacidades necesarias para conseguir una plataforma multiusuario.

### **2.2.2. Funciones de ensayo**

Las funciones de ensayo son los componentes necesarios para la realización de pruebas sobre los equipos bajo test, y que aportan la capacidad de estimulación y visualización de los datos y señales, programación de test y *recording* de datos para realizar un posterior análisis de los mismos. Están

embebidos en SEAS, permitiendo una correcta comunicación entre las diferentes funciones y facilitando una buena comprensión de las pruebas.

Estas funciones de ensayo están adecuadamente separadas, entendiendo cada función como un modulo independiente del resto, permitiendo que la evolución de cada uno pueda ser en paralelo con el resto.

#### **2.2.2.1. Graphical Data Visualization System (GDVS)**

GVDS [20] es un componente SEAS que permite al usuario monitorizar y estimular señales con la configuración de panel que elija.

Para una mejor comprensión se utiliza una interfaz gráfica, en la que para cada señal, el usuario puede elegir el tipo de representación que desee; indicadores, *switches*, *circuit breakers*, contadores, *sliders*, *displays*, gráficos, modelo pro defecto asociado al tipo de señal, etc.

El usuario puede crear sus propias clases gráficas y sus propios paneles, usando, si es necesario, editores comerciales. También puede crear grupos de paneles antes de realizar el test, de forma que permite preparar con antelación la información visual necesaria para un conjunto de señales dado.

Además se podrá emplear en tiempo real, permitiendo al usuario añadir o eliminar señales de la simulación mientras se realizan las pruebas.

Se presentan a continuación las capacidades de la herramienta:

- Visualización.
- Estimulación.
- Representación.
- Simulaciones Gráficas.

#### **2.2.2.2. Recording**

El entorno de gestión de pruebas de SEAS permite la capacidad de grabar el valor de conjuntos de señales durante la realización de test.

*Recording Definition Tool* [21] es una herramienta diseñada para especificar las señales y las velocidades de adquisición de las señales a grabar, permitiendo la posibilidad de analizar los datos a posteriori. Además otorga la capacidad de añadir señales de control al diccionario con el fin de inicializar comenzar y finalizar el *recording* de forma autónoma.

Para simplificar la adquisición de las señales, se genera un fichero binario con los valores de las señales, cuya lectura deberá realizarse a través de otra herramienta, *Analyser*.

Los pasos necesarios para el *recording* son los siguiente:

- Añadir señales.
- Configuración de señales.
- Ejecución del *recording*.

#### **2.2.2.3. Sequencer Executer**

Esta herramienta aporta a SEAS la capacidad de correr un proceso de señal que estimule un conjunto de señales en un tiempo determinado, y lo compare con valores de salidas esperados.

La definición de las secuencias se realizará a través de la herramienta *Sequence Definition* [22], que permite definir el conjunto de señales a estimular, y sus valores, y el conjunto de señales de salida con su salida esperada.

Permite realizar todas las definiciones necesarias independientemente unas de otras, y a la hora de ejecutarlas realizar una lista con los pasos que se van a seguir. Se podrá realizar secuencias de inicialización o de preparación de una situación concreta, ya que muchos de los sistemas a analizar tendrán memoria de su situación pasada.

Hay 2 posibilidades de ejecución:

I. Timed

Cuando se inicializa el proceso, se comienza la ejecución de la lista de secuencias definidas. El proceso empieza a seguir la lista paso a paso, indicando por pantalla el punto en que se encuentra: "RUNNING SEC = X". Cuando el servidor finalice el proceso, la herramienta comenzará a ejecutar la siguiente lista de secuencias.

El resultado final quedará almacenado en ficheros, que el usuario podrá recuperar para analizarlos.

II. Asynchronous

En éste modo la evolución es la misma, sólo que ahora se podrá manejar la evolución del proceso a través de la señal EVENT\_SEQUENCE.NEXT\_SEC. Con esta señal se indicará el paso de una secuencia a otra.

Igualmente el resultado quedará almacenado en ficheros para su posterior análisis.

**2.2.2.4. Análisis**

Existe una utilidad SEAS [23] para analizar los datos recogidos en simulaciones. Otorga la capacidad de análisis de las señales *off-line*. Facilita el acceso a todos los ejemplos, tiempos y valores de todas las señales recogidas.

Provee dos aplicaciones fundamentales:

- *Analyser*. Está diseñado para representar escrita o gráficamente los datos recogidos en simulaciones.

- *Recording Filtering Utility*. Esta aplicación transforma un fichero de recording en “.csv”, para su posterior análisis en SW comercial.

### **2.2.3. Simulaciones de SW específico**

SEAS *Simulations* [24] es un componente que permite una fácil integración de modelos/simulaciones, creadas por los usuarios, en los nodos SEAS. Los componentes necesarios para el uso de simulaciones están integrados en el *run time environment* para interoperar con el resto de los componentes. Estas simulaciones pueden ser controladas por los usuarios, pueden ser depuradas o integradas en el sistema.

Una simulación es un proceso cíclico que realiza operaciones de I/O sobre los valores de las señales. Se ejecutan con una política de programación de tareas basada en *frames*, en el procesador y a la tasa predefinida en ficheros externos. De esta forma varias simulaciones se pueden ejecutar al mismo tiempo.

El desarrollo de una simulación no se realiza conociendo el entorno SEAS. Lo que se necesita es crear una interfaz que contenga las funciones INIT, EXEC y STOP que inicialicen, ejecuten y paren la simulación respectivamente, y que permitan ejecutar la simulación paso a paso.

Los usuarios pueden crear sus propias simulaciones en FORTRAN, C, C++, ADA o cualquier lenguaje. El estándar está pensado para simulaciones ADA, pero también existe la posibilidad de crear y enlazar simulaciones en otro lenguaje.

El usuario también puede crear sus propios modelos usando otros entornos, como Matlab Simulink, tan sólo exportándolos en *Dynamic Link Library* (DLL) que podrán ser usadas por simulaciones. La integración entre las simulaciones y las señales definidas se realiza mediante ficheros “.xml”. Esta integración es suficientemente flexible para permitir que las simulaciones puedan relacionarse con cualquier señal.

#### **2.2.3.1. Binding Package**

Relacionar ADA con un lenguaje como por ejemplo C, requiere el uso de directivas de compilación para importar y/o exportar definiciones en cada lenguaje.

GNAT sigue el estándar llamada a las secuencias debidas, por lo que podrá relacionar con cualquier lenguaje que siga los estándares. Los siguientes son algunos de los identificadotes acordados:

- COBOL: Los valores deben ser pasados de acuerdo con las convenciones descritas en el apartado B.4 en el manual de referencia ADA 95.
- C: Los valores deben ser pasados de acuerdo con las convenciones descritas en el apartado B.3 en el manual de referencia ADA 95.
- Fortran: Los valores deben ser pasados de acuerdo con las convenciones descritas en el apartado B.5 en el manual de referencia ADA 95.
- StdCall: Es relevante en las implementaciones NT/Win95 de GNAT, y especifica que se debe usar como se indica en NT API.
- Dll: Se empleará lo mismo que para StdCall.

#### **2.2.4. Equipos necesarios y recursos**

A continuación se representa el esquema HW que tendremos en cualquier banco de ensayo. En los siguientes apartados se definirán la mayoría de apartados. Ahora definiremos los recursos necesarios para el subsistema de banco.



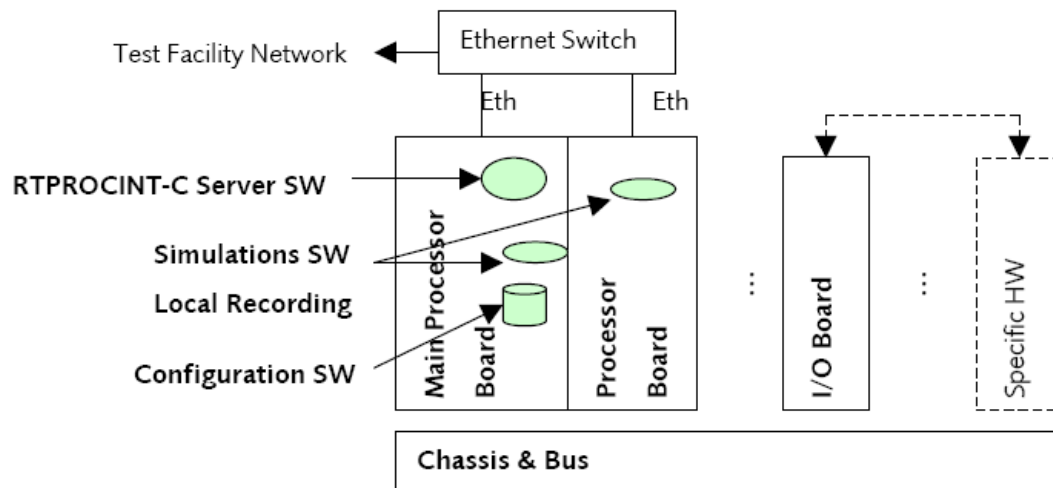


Figura 5 Representación de recursos [18]

El recurso principal sobre el que irá montado el SW de subsistema de ensayo será el procesador principal. Adicionalmente podremos disponer de una segunda tarjeta procesadora para cubrir las necesidades de procesamiento del sistema.

Los procesadores se comunican entre sí mediante Ethernet. Si es necesario más de un procesador para cubrir las necesidades básicas del SW se hará uso de un *switch* para conectar directamente los procesadores.

Los *Processor Boards* tendrán el SW específico y la configuración SW, mientras que el *Main Processor Board* tendrá el SW de subsistema de ensayo y configuración SW. Las simulaciones, *recordings*... pueden estar en cualquier *Processor Board*.

## 2.2.5. Tecnologías empleadas en el proceso de optimización

Un gestor de versiones es una herramienta que permite a un conjunto de usuarios mantener coordinados sus trabajos. Se suele emplear en entornos de programación donde se facilita la tarea de mantener la visión global del proyecto actual.

Se definen los SW de gestión de versiones presente en el mercado y que se emplearán en la optimización del subsistema de ensayo.

Existen dos productos en el mercado destinados a ésta tarea, CVS y Subversion.

(**Concepto de repositorio:** Un repositorio es simplemente un directorio en el servidor que contiene diversos módulos.)

#### 2.2.5.1. CVS

**C**oncurrent **V**ersioning **S**ystem (CVS) [12], [13] es un sistema de almacenamiento y organización de versiones a lo largo del tiempo para varios desarrolladores que deben trabajar concurrentemente, basado en licencia GPL. Está orientado a ficheros, asignando las versiones a cada fichero independientemente.

El uso del sistema se realiza generalmente mediante un servidor y un cliente, realizando las conexiones mediante un protocolo de seguridad, generalmente SSH.

El repositorio se encuentra en el servidor y el cliente tendrá una copia en directorio local para permitir la actualización de versiones en el servidor, y la actualización de la última versión en nuestro directorio local.

Al ser un sistema de reposición concurrente, lo habitual es su uso en proyectos en grupo, que requieren modificaciones de varios usuarios, por lo que se pueden producir conflictos.

- Commit (actualización de repositorio): Conflicto en *commit* sucede cuando un usuario trata de actualizar un fichero que ha sido

actualizado por otro usuario. Es fácil solucionarlo haciendo primero *update* del fichero en local y si no hay conflicto, realizar el *commit*.

- Update (actualización de local): Conflicto en *update* se produce si se intenta actualizar el local con un fichero de repositorio y ambos han sido modificados en la misma línea. En este punto la única solución es humana, y deberá resolverse usuario con usuario.

CVS permite a su vez realizar diferentes ramas en el repositorio, de un mismo proyecto, lo que permitirá mantener una rama principal, y diferenciar versiones que haya que particularizar para un cliente con exigencias concretas.

#### **2.2.5.2. Subversion**

**SubVersion** (SVN) [12] es un sistema de gestión de versiones de código abierto. El desarrollo está siendo realizado por CollabNet. Fue diseñado para mejorar las utilidades de CVS, pero facilitando la migración desde éste, ya que apenas se modifican los comandos.

Se sigue exactamente igual la filosofía servidor-cliente, situando el repositorio en servidor, con almacenamiento Berkeley DB, y manteniendo una copia en el cliente. Para realizar la conexión cliente a servidor se pueden utilizar dos protocolos:

- Svnserve: protocolo propio de SVN.
- WebDAV: utilizando además servidor Apache 2.x, que permite un mejor control de la seguridad del servidor.

A diferencia de CVS, SVN está basado en proyectos y no en ficheros, por lo que se mantiene la versión del conjunto de ficheros, y no individualmente.

Los conflictos a la hora de actualizar el repositorio o el directorio copia en local, son los mismos que en CVS, y la solución es la misma, copiar-modificar-actualizar.

Uno de los puntos más importantes de SVN es la completa API de que dispone para poder realizar aplicaciones visuales y de enlace hacia el código de SVN, permitiendo a los programadores y desarrolladores de código mayor facilidad de utilización e integración de SVN.

#### **2.2.5.3. Apache 2.x**

Apache es un servidor *web* de SW libre para distintos **Sistemas Operativos (SO)** (Windows, UNIX, Mac...) que implementa el protocolo *http/1.1*. Tiene una característica modular propia del SW libre, que permite añadir y modificar toda la funcionalidad que ofrece.

El servidor Apache se desarrolla dentro del proyecto *HTTP Server* (httpd) de la Apache Software Foundation.

Las características más importantes son:

- **Multiplataforma:** aunque idealmente está preparado para funcionar bajo UNIX.
- **Fácil configuración.**
- **SW Libre.**
- Útil para proveedores de Servicios de Internet que requieren **sitios pequeños** con páginas estáticas.
- Ámplias librerías de **PHP y Perl** a disposición de los programadores.
- **Modularidad.**
- Es capaz de utilizar lenguajes como **PHP, TCL, Python**, etc.

#### **2.3. Módulo de Comunicaciones**

El sistema de comunicaciones de un banco de ensayo involucra todos los intercambios de datos entre equipos propios de aviónica y equipos específicos del banco de ensayo. Habrá diversas partes interrelacionadas en las comunicaciones de un banco.

- Chasis
- Tarjetas de Comunicaciones
- Buses de datos de aviónica (*Avionic Buses*)
- Sensores y Actuadores

### 2.3.1. Chasis

Se quiere tener siempre un sistema abierto y flexible, por lo que se requerirá el empleo de buses de comunicaciones modulares.

Actualmente la mayoría de bancos están basados en dos buses de comunicaciones modulares:

- VME
- PXI

Otros buses de comunicaciones como son GPIB o Ethernet, no se basan en una estructura modular de los equipos de instrumentación, si no que permiten la interconexión de equipos ya existentes. Sin embargo el objetivo de los buses modulares es evitar la interacción directa del usuario sobre los equipos de instrumentación. Se tendrá un chasis sobre el que irán montados los diferentes equipos, y mediante comunicaciones a través del bus específico se podrán realizar las tareas de adquisición, almacenamiento y tratamiento.

#### 2.3.1.1. VME

VERSA Module Eurocard (VME) [10], [25] fue definido en 1981 y estandarizado en 1987 (IEEE 1014) por un consorcio de empresas (Motorola, Mostek y Signestics) con el objetivo de crear un sistema modular y de propósito general para adaptar diferentes tipos de procesadores existentes.

Las características principales son:

- **Abierto:** Hay un estándar internacional pero cualquiera puede modificar las normas para adecuarlo a sus necesidades.
- **Modular:** Se definen 21 *slots* para tarjetas y procesadores, pudiendo ampliar la configuración añadiendo nuevos chasis en serie o paralelo.
- Compatible con otras tecnologías.
- **Asynchronous:** No limitado por una sincronización central.
- **Tiempo Real:** Determinista y estructurado con jerarquía de prioridades.
- **Multiproceso:** Los procesos están separados y distribuidos.
- Independiente de la tecnología. Todos los  $\mu$ Procesadores se pueden usar.
- Configurable dinámicamente.
- Velocidad de 320Mbps.
- Configuración Maestro-Esclavo

Define una estructura modular en la que el slot 1 será una tarjeta controladora y el resto podrán ser procesadoras, tarjetas de memoria, e/s...

Los siguientes buses de comunicaciones serán los que se encuentran integrados en el chasis:

- **Bus de transferencia de datos:** Asíncrono y controlado por el dispositivo maestro. Organizado con un bus de datos, bus de direcciones, líneas modificadoras de dirección y líneas de control.
- **Bus de interrupciones priorizadas:** Peticiones de interrupciones y tratamiento de actividades programadas que requieren el uso inmediato.
- **Bus de arbitración:** Control del acceso al bus de transferencia mediante líneas de peticiones, liberaciones...
- **Bus de utilidades:** Mediante líneas se definen utilidades de reloj, sincronización, fallos en tensión, inicialización...

Con VME se tiene la capacidad de crecer tanto en tamaño como en velocidad de procesamiento sin necesidad de rediseñar ni volver atrás en el trabajo ya realizado.

Se emplearán diferentes buses para el núcleo de VME y se podrá disponer de otros *sub-buses* para comunicaciones.

Core Buses:

- VME
- VME64
- VME64X
- 2eVME
- etc...

Sub-Buses:

- Ethernet
- RapidIO
- PCI express
- etc...

Year	Core Technol.	Interconnects & Fabrics		Mezzanines		Support & Technical Recommend.	
2005			VITA 49			ANSI/MTA 47	VITA 52
			VITA 46			VITA 48 & 50	VITA 51
	ANSI/MTA 1.5	ANSI/MTA 31.1	ANSI/MTA 17.1	ANSI/MTA 39	ANSI/MTA 32	ANSI/MTA 38/40	ANSI/MTA 1.7
		VITA 41				ANSI/MTA 30.1	
2000	VITA 34			ANSI/MTA 29	ANSI/MTA 20	VITA 37	
				ANSI/MTA 35	VITA 36	ANSI/MTA 30	ANSI/MTA 1.6
		ANSI/MTA 5.1					
	ANSI/MTA 23	ANSI/MTA 26	ANSI/MTA 17				ANSI/MTA 23
	ANSI/MTA 1.1			ANSI/MTA 4.1	ANSI/MTA 12	ANSI/MTA 1.3	ANSI/MTA 25
		ANSI/MTA 6.1					
1995		ANSI/MTA 13	ANSI/MTA 10	ANSI/MTA 4		ANSI/MTA 3	
	ANSI/MTA 1	ANSI/MTA 6					
1990							
	IEEE 1014						
	IEC 821	VSB					
1985	VME Rev. C	VMX					
		VMX					
	VME Rev. B						
	VME Rev. A		VMS				
1980							
	Connectors	3-row DIN	5-row DIN	2 mm	MultiGig (V41)	MultiGig RT (V46)	
	Mechanical	EuroCards	19" racks	IEEE 1101.1/2	IEEE 1101.10/11		

**Figura 6 Características VME**

### 2.3.1.2. PXI

El bus **PCI eXtension for Instrumentation (PXI)** [26], [27] es un bus industrial de comunicaciones estándar para instrumentación y control. Las siglas significan una extensión del bus PCI pensada para aplicaciones de instrumentación. Fue diseñado por National Instruments en 1997.

PXI aprovecha la extensión en el mercado del estándar PCI para introducir mejoras y que los usuarios puedan emplear las mismas herramientas y sistemas existentes pero con mayor funcionalidad.

Las características fundamentales del estándar son:

- Basado en PCI para sistemas de automatización y medición.
- Combina características del bus eléctrico PCI con la modularidad de Eurocard del CompactPCI.
- Añade sincronización especial de buses y características especiales de software.

Los sistemas PXI constan de 3 partes:



- Chasis
- Controlador
- Módulos

El chasis proporciona la característica modular para el sistema. Generalmente tiene un rango de tamaño de 4 a 18 ranuras, y se encuentran también disponibles con características especiales como una fuente de DC y acondicionamiento de señales integrado.

El chasis contiene un plano trasero de alto desempeño para el PXI, que incluye el bus PCI y buses de sincronización y disparo. Dichos buses de sincronización y disparo permiten a los usuarios desarrollar sistemas para aplicaciones que requieren una sincronización de precisión. A la hora de tomar señales todas ellas estarán referidas a la misma base de tiempos, garantizando la integridad de datos.

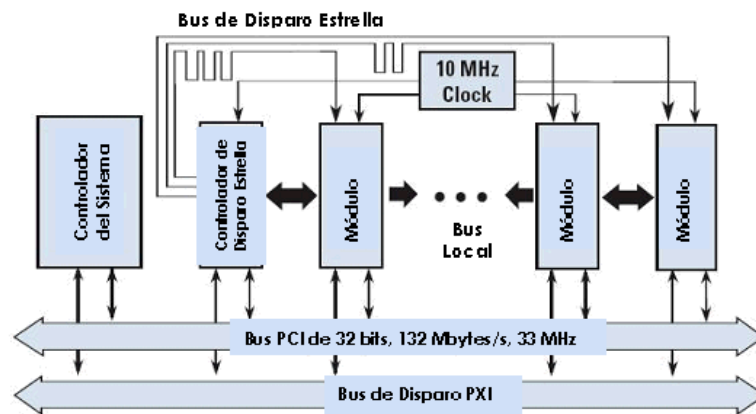


Figura 7 Buses de sincronización y disparo [26]

PXI es directamente compatible con CompactPCI, así que cualquier módulo CompactPCI 3U puede utilizarse en un sistema PXI. Adicionalmente, las tarjetas bus y tarjetas PMC (**P**CI **M**ezzanine **C**ard) pueden ser instaladas en sistemas PXI utilizando módulos adaptadores.

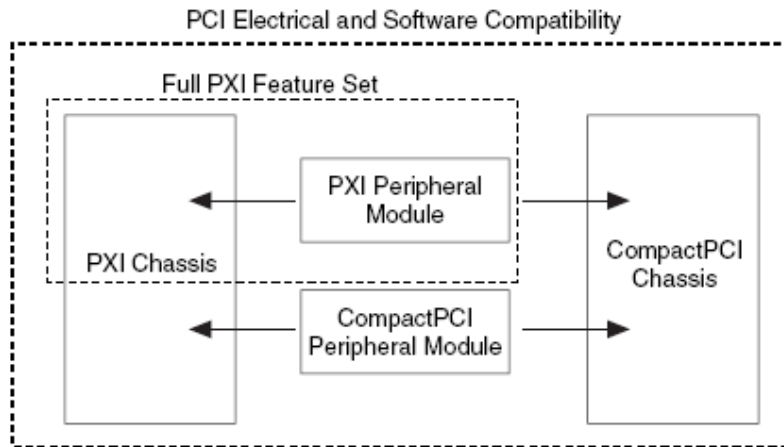


Figura 8 Integración PXI y cPCI [27]

Características eléctricas y de conexión de tarjetas:

- Disponibilidad de 33/66 MHz. (7 periféricos con 33MHz, y 4 con 66MHz).
- Transmisión de datos en 32-bit y 64-bit.
- Tasas de transmisión de 132 Mbps (32-bit, 33MHz) a 528 Mbps (64-bit, 66 MHz).
- Capacidad de expansión del chasis mediante PCI-PCI bridges.
- Migración hacia 3.3V.
- Capacidad Plug and Play.

### 2.3.2. Buses de Comunicaciones en aeronáutica

En aeronáutica las comunicaciones y transmisiones de señales dentro del avión se realizan con tecnologías propias del sector. No se emplean buses comerciales debida a las necesidades de seguridad, redundancia y velocidades.

En los bancos de ensayo, por lo tanto, hay que emplear estas tecnologías para representar fielmente los sistemas de avión. Sin embargo para las propias comunicaciones del banco, se emplean tecnologías comerciales que se ajusten mejor a las necesidades.

Se emplearán además otros medios de transmisión digital y analógica para señales de control, monitorización, visualización... Estas transmisiones estarán sometidas a especificaciones concretas, dependiendo del estándar que se elija para el banco. Estas señales, tanto discretas como analógicas estarán conectadas directamente a una entrada de una tarjeta del banco, y servirán como control de alguna función como encendido y apagado de algún módulo...

Entre los estándares de comunicaciones [38] que se usan en aviónica, los más comunes son los siguientes:

- Arinc 429
- AFDX
- MilStd 1553
- Stanag 3910
- CAN

También se emplearán dos canales de transmisión de señales analógicas y digitales, no específicos, que emplearán diferentes tipos de tarjetas.

- Digital Input Output (DIO)
- Analog Input Output (AIO)

#### 2.3.2.1. **ARINC429**

**Aeronautical Radio, Inc.** (ARINC) es una empresa fundada en 1929 por la Federal Radio Comision, para coordinar y aportar un estándar único de comunicaciones radio fuera del ámbito estatal. Sólo empresas relacionadas con la aviónica y compañías aéreas pueden ser accionistas, aunque cualquier empresa de aviación podrá usar los productos ARINC.

ARINC429 [28], [29] fue desarrollado y es gestionado por Airlines Electronic Engineering Comité (AEEC) junto con representantes de compañías aéreas, gobierno y ARINC. La especificación de ARINC429 define las necesidades para un sistema de transmisión de datos basado en una sola fuente de información y muchos receptores (hasta 20). El máximo número de receptores estará limitado por la mínima impedancia de entrada de los receptores.

Cada avión debe estar equipado con gran cantidad de equipos electrónicos y sistemas que deberán estar conectados con ARINC429. La especificación define exhaustivamente todos los detalles de instalación e implementación de las diferentes aplicaciones.

Eq. ID	Equipment Type	Eq. ID	Equipment Type
001	Flight Control Computer (701)	029	ADDCS (729) and EICAS
002	Flight Management Computer (702)	02A	Thrust Management Computer
003	Thrust Control Computer (703)	02B	Perf. Nav. Computer System (Boeing 737)
004	Inertial Reference System (704)	02C	Digital Fuel Gauging System (A310)
005	Attitude and Heading Ref. System (705)	02D	EPR Indicator (Boeing 757)
006	Air Data system (706)	02E	Land Rollout CU/Landing C & LU
007	Radio Altimeter (707)	02F	Full Authority EEC-A
008	Airborne Weather Radar (708)	030	Airborne Separation Assurance System
009	Airborne DME (709)	031	Chronometer (731)
00A	FAC (A310)	032	Passenger Entertain. Tape Reproducer (732)
00B	Global Positioning System	033	Propulsion Multiplexer (PMUX) (733)
00D	AIDS Data Management System	034	Fault Isolation and Detection System (734)
010	Airborne ILS Receiver (710)	035	TCAS (735)
011	Airborne VOR Receiver (711)	036	Radio Management System (736)
012	Airborne ADF System (712)	037	Weight and Balance System (737)
016	Airborne VHF COM Receiver (716)	038	ADIRS (738)
017	DEFDARS-AIDS (717)	039	MCDU (739)
018	ATC Transponder (718)	03A	Propulsion Discrete Interface Unit
019	Airborne HF/SSB System (719)	03B	Autopilot Buffer Unit
01A	Electronic Supervisory Control	03C	Tire Pressure Monitoring System
01B	Digital Flap/Slat Computer (A310)	03D	Airborne Vibration Monitor (737/757/767)
01C	Engine Parameter Digitizer (Engine)	03E	Center of Gravity Control Computer
01D	A/P & F/D Mode Control Panel -757/767	03F	Full Authority EEC-B
01E	Performance Data Computer (Boeing)	040	Cockpit Printer
01F	Fuel Quantity Totalizer	041	Satellite Data Unit
020	DFS System (720)	046	CTU
023	Ground Proximity Warning Sys (723)	047	Digital Flight Data Recorder
024	ACARS (724)	----	additional items
025	Electronic Flt. Instruments (725)	----	"
026	Flight Warning Computer (726)	----	"
027	Microwave Landing System (727)	241	High Power Amplifier

Figura 9 Tipos de equipos ARINC429 [28]

ARINC429 es uno de los estándares más ampliamente utilizado en aviónica actualmente. Existen otros estándares en progreso, pero sigue

manteniendo una hegemonía difícilmente sustituible en sistemas de comunicaciones de aviónica.

#### I. ¿Qué es ARINC429?

ARINC429 es un estándar que define cómo deben comunicarse los sistemas y equipos de comunicaciones de avión. Se definen las características eléctricas, señales, protocolos e interconexión. Emplea Mark 33 **D**igital **I**nformation **T**ransfer **S**ystem (DITS) basado en un bus de datos unidireccional.

Define la separación física de los canales de envío y recepción para cada sistema de avión, por lo que emplea mucho cableado. Aunque se especifica, igualmente, la cantidad de quipos que pueden estar interconectados para la recepción de datos.

Se establece un tipo de topología *broadcast* con un equipo (LRU) transmitiendo y varios recibiendo. Por lo que se definirán tiempos suficientes de separación de envíos para detección de colisiones y errores.

#### II. Capa física

ARINC429 usa par trenzado apantallado para la transmisión de las palabras de datos (32 bit). La transmisión se realiza de forma diferencial entre cada cable del par trenzado. De los dos cables, uno se especifica como HI y otro como LO haciendo uso de la modulación **BiPolar Return to Zero** (BPRZ).

##### **2.3.2.2. AFDX**

**Avionic Full Duplex Switched Ethernet (AFDX)** [30], [31], [32] Network es una red aviónica determinista basada en la tecnología 10/100Mbps Full Duplex Switched Ethernet. AFDX es evolución de ARINC429 y aunque en aviónica hoy en día siguen conviviendo, es el sistema de comunicaciones más empleado en modelos como el A400M o A330.

AFDX es parte de la especificación ARINC 664, que define como las tecnologías de redes comerciales (**Comercial Off-The-Shelf COTS**) se deben usar en las futuras redes de aviónica (ADN). AFDX es una definición de AIRBUS.

En ARINC 664 se define tanto la capa física como el protocolo de comunicaciones entre los **End-Systems** (ES) o equipos de aviónica finales de la red, por lo que AFDX podrá emplear cualquier capa física aquí definida. Cada equipo 2 puertos de salida, cada uno con 2 pares trenzados, teniendo redundancia total de la red y capacidad *full-duplex* completa.

AFDX es determinista. Predice el estado de la red, el retardo y la temporización de las comunicaciones.

#### I. Capa Física. Switched Ethernet

En aviónica no se permite la incertidumbre de colisión de Ethernet, por lo que para conseguir determinismo, entre otros aspectos, se usa *switched ethernet*.

Esta tecnología se basa sustituir el medio físico compartido por un canal dedicado para cada ES. Este canal conecta cada ES con un *switch*. Cuando llega un *frame* analiza la dirección MAC destino y dependiendo de la tabla la envía por la interfaz adecuada. Cada canal además será *full-dúplex*.

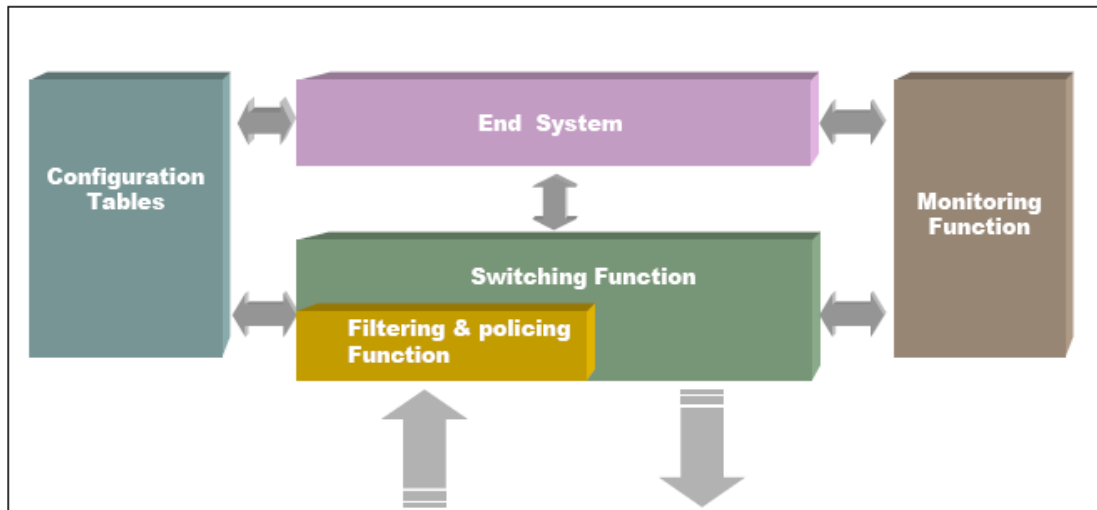


Figura 10 Esquema Switched Ethernet AFDX [31]

El proceso de *filtering&policy* se explicará más adelante. Este depende del *VirtualLink*, parámetros de retardo, *Bag...*<sup>2</sup>

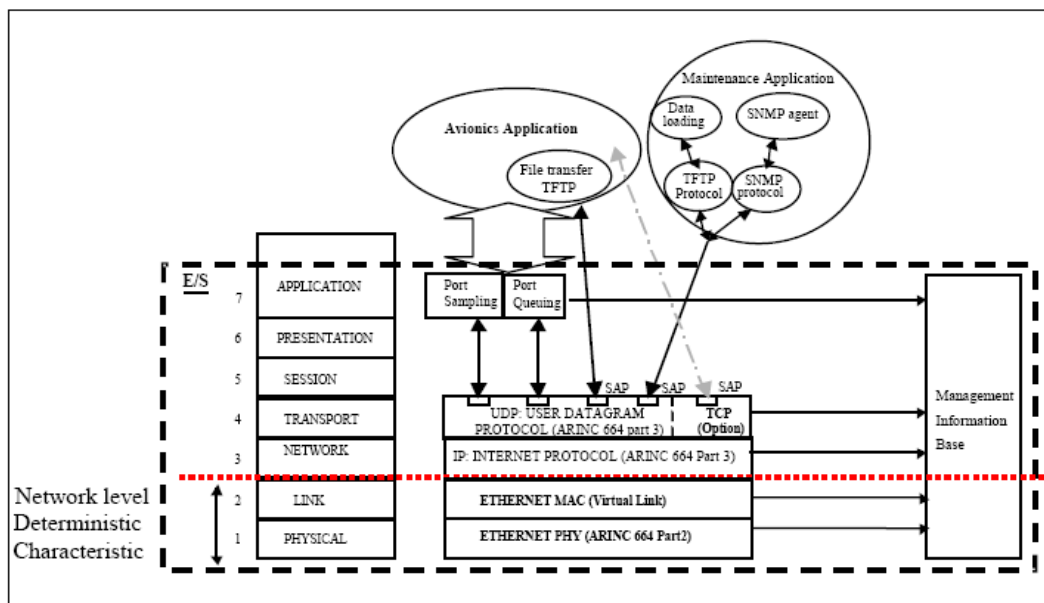


Figura 11 Configuración AFDX [30]

Con esta configuración se soluciona el problema de colisiones, pero aparece un nuevo problema. Ahora es necesario modelar cada *switch* de acuerdo con las capacidades de la red.

<sup>2</sup> Conceptos explicados en Anexo.

### 2.3.2.3. MILSTD 1553

A mediados de los 60's, comenzó a crecer la complejidad en los sistemas de aviónica. Y como consecuencia creció el número de señales discretas necesarias, lo que implicaba crecimiento de tamaño y peso. Para solucionar este problema, la Aerospace Branch of the Society Automotive Engineers creó el primer esquema de un bus estándar de datos multiplexados (MIL-STD-1553), que se usará en servicios militares.

MIL1553 [33], [34], [35] **sentó las bases del time-division multiplexing**, enviando datos desde múltiples equipos de aviónica a través de un único par trenzado. El estándar se sigue utilizando en aviones militares, helicópteros, *space systems* y vehículos de tierra.

#### I. Tipos de Terminales

- **Bus Controller (BC):** Transmite y recibe datos y controla el flujo de datos en el bus. Toda la información se intercambia en modo command/response. Es el encargado de iniciar cualquier transmisión de datos. Será capaz de responder a los cambios de estado del avión, como por ejemplo pasar de tierra a aire, o de modo normal a combate.
- **Remote Terminal (RT):** Este tipo de equipo recibe información y la transforma para su envío por el bus. También puede recibir información por el bus de datos y transformarla para otra aplicación. Esta situado entre un equipo de aviónica y el bus de datos. No envía información a menos que se lo indique el BC. Puede haber hasta 32 equipos de aviónica.



- **Bus Monitor (BM):** Es un equipo únicamente de test. Monitorea y graba toda la información, o la que se especifique, del bus de datos para su posterior análisis.

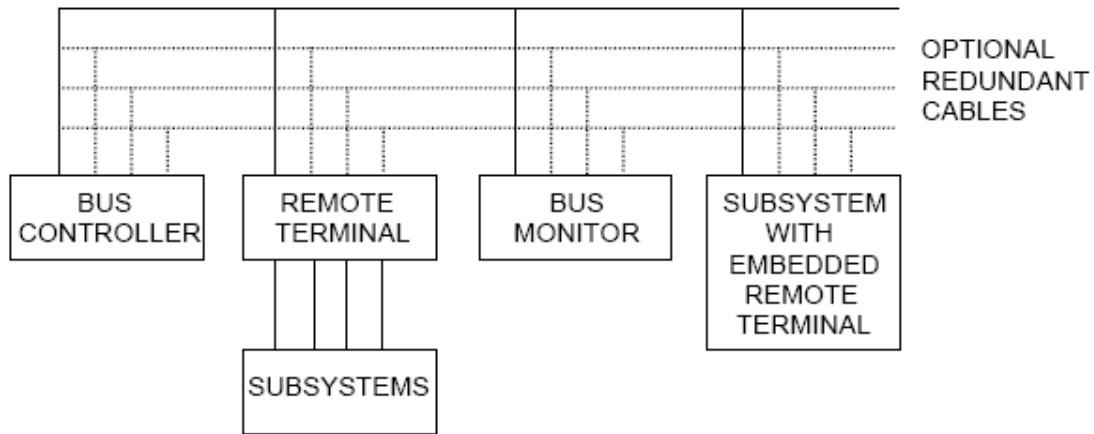


Figura 12 Esquema MIL1553 [33]

#### 2.3.2.4. STANAG 3910

STANAG 3910 [36] es un protocolo empleado a partir de 1989 que será la evolución de 1553 para utilizar un bus de más velocidad. De hecho la mejora consiste en emplear la estructura de MIL1553 y añadir funcionalidad mediante otro bus de comunicaciones.

La topología empleada por 3910 será:

- Bus 1553 para control y comandos (1Mbps) (LS)
- Bus Fibra Óptica para envío de datos a más velocidad (20Mbps) (HS)

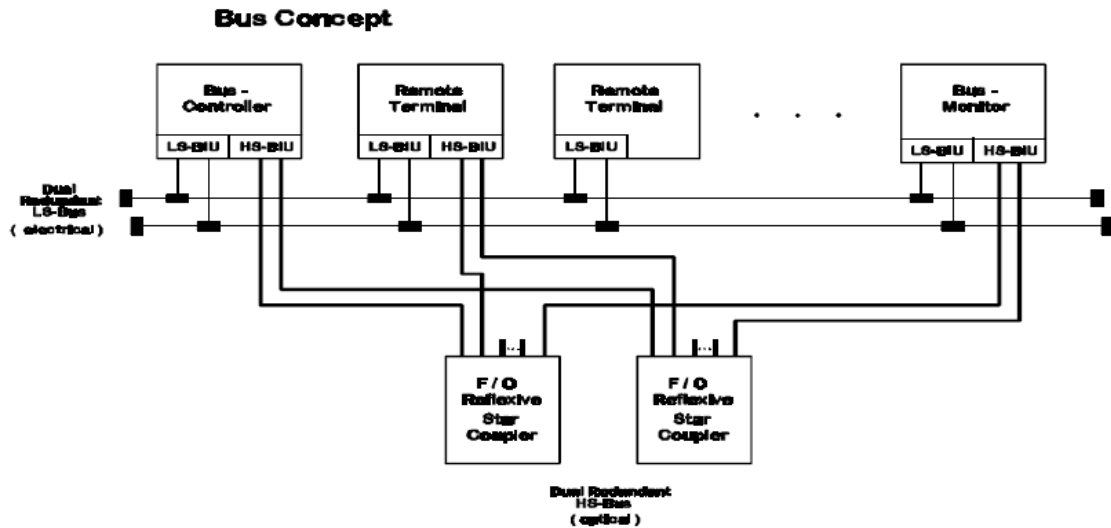


Figura 13 Bus Concept 3910 [36]

#### 2.3.2.5. CAN

La especificación del bus **C**ontroller **A**rea **N**etwork (CAN) [37] sólo implementa las dos capas inferiores de la pila de protocolos OSI, capa física y la capa de enlace. Tiene dos niveles con 1000Kbps y 125Kbps.

CAN es un protocolo basado en mensajes no en direcciones. Emplea broadcast, cualquiera recibe el mensaje y cada uno decide si emplearlo o no.

Es un protocolo ampliamente empleado en las comunicaciones en automóviles. Es un sistema diseñado para su uso en modo maestro esclavo, lo que permite una estructura centralizada, en la que la información sólo fluye desde los puntos que se exige. En caso de emergencias no se pierden recursos, si no que se exige la información a aquellos equipos que en ese momento son más importantes.

Es menos empleado en aviónica debido a la baja velocidad y las restricciones maestro-esclavo en un entorno con tantos componentes y recursos. Aunque ofrece otras posibilidades de comunicación que permiten adaptarlo a las necesidades de la aeronáutica.

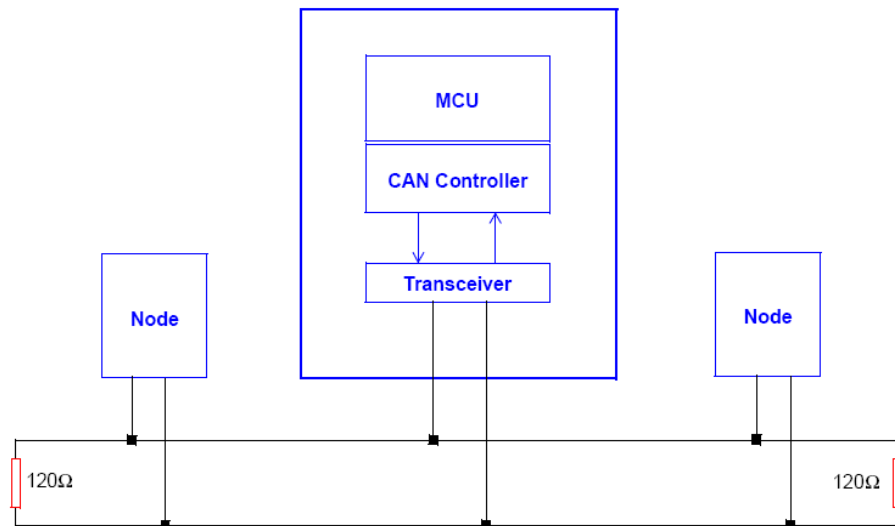


Figura 14 Topología CAN [37]

El uso del bus se puede realizar de diferentes modos, Maestro/Esclavo, Multimaestro, *Bit-wise arbitration*, *Daisy Chain*...

#### 2.3.2.6. DIO

Casi todos los bancos de pruebas de equipo, subsistemas y sistemas, necesitarán una interacción directa con personal humano de 2 de formas:

- A través de las aplicaciones clientes del subsistema de ensayo.
- Directamente.

Para esta última forma de interacción se diseñan los *patching panels*, de acuerdo con las necesidades del cliente, y teniendo en cuenta las propiedades del banco.

Un *patching panel* es un mosaico de *switches* claramente identificados que aportan la utilidad necesaria al banco. Generalmente, al ser *switches* tendrán posibilidades discretas, es decir, apagar o encender; apagar, encender o reposo; esclavo o maestro; activo o pasivo... que indicarán un valor al

subsistema de ensayo a través de la tarjeta de discretas, y éste actuará en consecuencia sobre el equipo concreto.

En un banco de comunicaciones, como ejemplo, puede ser necesario activar un canal de radio por el que se va a transmitir, activar la escucha de la radio...

Estas señales estarán sujetas al formato de entrada de la tarjeta elegida, como puede ser OpenGND, OPENVext, 28-V/Open, 28-V/GND, TTL...

#### **2.3.2.7. AIO**

Las transmisiones analógicas serán necesarias para la adquisición de información continua. Estas señales en general se obtendrán desde los múltiples sensores disponibles, aunque como casos particulares están las transmisiones de video, de audio, voz...

La transmisión de estas señales se acota por las especificaciones de la tarjeta analógica de adquisición de señales, que generalmente define el rango de valores, +12V a -12V, +10V a -10V...

Las tarjetas de adquisición de señales tendrán especificaciones de entrada de las señales analógicas, y unas especificaciones de valores de salida. En general, las tarjetas actuarán como conversor analógico digital.

#### **2.3.3. Tarjetas de Comunicaciones**

Las tarjetas de comunicaciones son una parte fundamental del banco de ensayo. En el esquema del banco se tiene un equipo de avión con diferentes señales de salida y entrada a través de los buses de aviónica específicos y por otro lado un subsistema de banco de ensayo que necesita obtener y estimular estas señales. Para ello se hace uso de las tarjetas de comunicaciones.

Una tarjeta de comunicaciones es un elemento de adaptación de información entre dos elementos del banco. De una parte está un bus de comunicaciones (AIO, DIO, ARINC429...) y del otro se puede tener otro bus de comunicaciones (ETHERNET, USB...) o directamente un chasis VME sobre el que se transmitirá la información necesaria.

I. AMC 1553

La tarjeta AMC1553 [39] de AIM está rediseñada para analizar, simular, monitorizar y probar sistemas de aviónica basados en el bus de datos MIL-STD-1553. La interfaz de tarjeta implementa IO para cada canal e IRIG IN/OUT.

II. AMC 429

AMC429 es una tarjeta de AIM [40] diseñada para simular, analizar, monitorizar y probar sistemas de aviónica basados en el bus ARINC429.

AIM provee un modulo PMC que permite el montaje sobre tarjetas de adaptación a PCI, VME o CPCI, además de tener la capacidad de conectarse a un servidor integrado con puertos PMC.

III. AVI 3910

AVI3910 [41] es una nueva tarjeta perteneciente a los módulos de VME-Bus, empleada para analizar, monitorizar y probar sistemas de aviónica basados en el bus STANAG3910.

Se emplea esta tarjeta para simulación, monitorización y estimulación de errores en el bus. Con esta tarjeta se tienen dos canales 1553 y dos canales 3910 sobre un mismo *slot* VME. El enlace de Fibra Óptica (FO) de 3910 está implementado en otro módulo, VOX3910, que proporciona la capacidad de conexión por FO necesaria en 3910.

#### IV. PCI-CAN/XS2

La tarjeta PCI-CAN/XS2 de National Instruments (NI) [42] es una interfaz CAN de 2 puertos seleccionable por transceptor diseñada para comunicar con dispositivos CAN de alta velocidad, de baja velocidad o de un solo cable.

El transceptor adecuado para cada puerto es seleccionado por software a través de una aplicación ofrecida por NI. Los módulos NI CAN para PCI utilizan el Controlador Philips SJA1000 CAN para funcionalidad avanzada como escuchar únicamente, recepción personal (echo) y modos avanzados de filtrado y nuevos transceptores para modo sleep/wakeup.

##### **2.3.4. Sensores y Actuadores**

En bancos de ensayo basados en tierra, generalmente, no se emplearán sensores reales de avión, ya que no tiene sentido tener montado un sistema GPS, altímetro o antena de comunicaciones.

Sin embargo, en bancos HW de integración de subsistemas, debe estar montado el máximo de componentes reales como sea posible. Por lo tanto, para probar la integración de subsistemas estará al máximo detalle la cabina de piloto del avión incluyendo todos los actuadores propios como son *joysticks*, *circuit breaker*, señales discretas...

Estos actuadores serán particulares y propios de cada diseño, por lo que realizar un estudio de estos elementos no tiene sentido más allá de conocer su uso y sentido dentro del avión.

En general, los actuadores son equipos diseñados para interaccionar manualmente con los equipos de avión con el objetivo de modificar alguna de las condiciones actuales en las que se encuentra. La forma de interaccionar será distinta dependiendo del avión, desde acción directa sobre algún equipo o sistema hasta un modelo basado en SW previo de tratamiento de la acción a tomar.

Actualmente se trata de simplificar las actividades del piloto reduciendo al máximo sus responsabilidades y decisiones, permitiendo al piloto estar concentrado en las necesidades primarias.

### **2.3.5. Tecnologías empleadas en el proceso de optimización**

Para el proceso de optimización del modulo de comunicaciones se hará uso de matrices de conmutación. Para esto es necesario conocer las tecnologías disponibles para la nueva estructura.

#### **2.3.5.1. Conmutación**

La conmutación [1] es la capacidad de conexión entre dos usuarios de un sistema mediante la configuración de una red intermedia de la que ambos usuarios son miembros.

Existen 2 tipos fundamentales:

##### **I. Circuitos**

La conmutación de circuitos implica la creación de un canal de comunicación dedicado entre ambos usuarios. Se siguen tres fases: establecimiento del circuito, transmisión de información y desconexión del circuito.

Ha sido típicamente el método empleado en transmisiones telefónicas, puesto que ambos usuarios precisaban de una conexión permanente. Sin embargo, el rendimiento suele ser bastante bajo, ya que el canal esta reservado permanentemente y el uso no suele ser permanente en ambas direcciones.

La parte importante es tener un retardo completamente determinista en caso de producirse el establecimiento de canal, que dependerá del uso de la red.

## II. Paquetes

Como mejora para aplicaciones en las que las necesidades de transmisión son a ráfagas, en las que no existe la necesidad de tener un canal establecido permanentemente, surge la conmutación de paquetes.

No existirá un canal dedicado entre ambos usuarios, si no que se envía la información en paquetes de tamaño limitado, a través de la red. Estos paquetes contienen la dirección de destino, y será cada nodo intermedio de la red, el que se encargue de encaminarlo adecuadamente.

### 2.3.5.2. Matriz de Conmutación

El término de **matriz de conmutación** está altamente ligado a la telefonía. Son los elementos empleados para encaminar las conexiones telefónicas entre las diferentes líneas, aunque se ha extendido el concepto más allá de la telefonía. Actualmente la matriz de conmutación interconecta puntos con puntos según las necesidades del usuario.

El primer paso para estudiar la evolución y situación de las matrices de conmutación es comprender lo que es una red de conexión. Una **red de conexión** es el HW empleado para establecer los circuitos físicos entre 2 puntos de conexión. En orden de antigüedad, las tecnologías empleadas son las siguientes:

- Manual
- Electromecánica
- Electrónica
- Óptica



### 2.3.5.3. Evolución de comunicaciones en instrumentación

La instrumentación es el grupo de equipos y elementos utilizados en el sistema para medir y registrar valores de variables, con el fin de comprobar y mejorar el funcionamiento de éste.

La evolución hacia las últimas novedades en instrumentación obliga a realizar un pequeño trayecto para analizar el mercado actual.

#### I. Buses de comunicación independientes

Estas tecnologías físicas por las que se ha movido el mercado a lo largo del tiempo están diseñadas para equipos de instrumentación independientes de las comunicaciones. En orden de aparición las tecnologías son las siguientes:

##### a. GPIB (1965)

En 1965, Hewlett-Packard diseñó el **Hewlett-Packard Interface Bus** (HP-IB) para interconectar los equipos programables a los ordenadores, con el fin de tener mayor accesibilidad. Gracias a las tasas de transferencia tan altas conseguidas (8Mbps posteriormente ampliado a 64Mbps), el protocolo fue adquiriendo gran popularidad en instrumentación, pasando a ser el estándar IEEE-488. Actualmente es más común el nombre **General Purpose Instrument Bus** (GPIB) [2].

La estructura del bus se basa en *talkers*, *listeners* y *controllers*.

- **Talkers:** Envía datos.
- **Listeners:** Recibe datos.
- **Controllers:** Controla el flujo enviando comandos a los dispositivos.

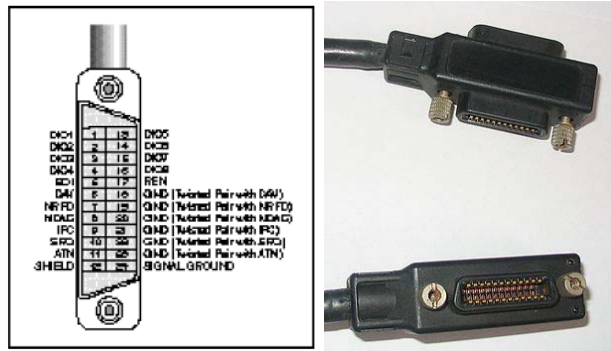


Figura 15 Conector GPIB [43] [44]

Es un bus de datos paralelo que permite configuraciones tanto línea como estrella, pero con restricciones sobre la longitud y cantidad de dispositivos.

- Separación de 4m entre equipos y 2m con respecto al bus.
- Longitud del cable 20m.
- Máximo de 15 dispositivos por bus.

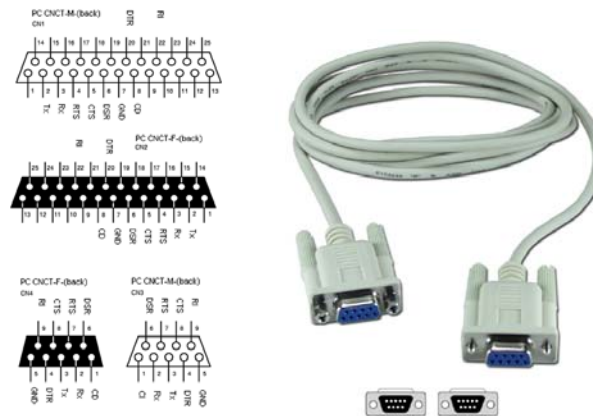
Las siguientes evoluciones del estándar fueron:

- IEEE 488.1, define HW.
- IEEE 488.2, define sintaxis.
- IEEE 488.1-2003, aumenta velocidad a 8Mbps.

b. RS-232 (1969)

Bus de transmisión serie para intercambio binario entre equipo terminal de datos (DTE) y equipo de comunicación de datos (DCE). El estándar surge a partir de un proyecto de Electronic Industries Association, en el que se pretendía comunicar equipos mediante línea de teléfono, empleando módem en cada punto. El modelo era simple pero se producían múltiples errores, por lo que se definió el estándar RS-232 [3], [4] en el que se definía la temporización, señales...

A pesar de existir múltiples opciones, se permite la comunicación únicamente con 2 cables, pudiendo realizar el envío en 7, 8 ó 9 bits. Normalmente se transmiten datos a 9600 baudios (bps), aunque se permiten velocidades superiores pero se reducirá la longitud permitida del cable.



**Figura 16 Conector RS-232 [45], [46]**

La popularidad del protocolo provocó la aparición de múltiples mejoras:

- RS-232-C
- V.24
- RS449, descripción mecánica.
- RS423, descripción eléctrica.

Y otras evoluciones superiores como son:

- RS422
- RS485

Es un protocolo empleado para conexión de equipos de instrumentación a otros equipos donde recoger y manipular los datos, pero no suele ser empleado para adquirir datos.

c. Ethernet (1975)

Es un estándar de redes de área local basado en acceso al medio mediante contienda, **Carrier Sense Multiple Acces/Collision Detection** (CSMA/CD), característica principal del protocolo. Mediante esta técnica de acceso se aprovecha el canal en mayor medida que con Aloha o Aloha Ranurado.

Ethernet [5] es popular porque permite un buen equilibrio entre velocidad, costo y facilidad de instalación y soporta todos los protocolos de red más populares. La topología más común es la de estrella (bus en topologías antiguas) permitiendo el uso de *hubs* y *switches* (estrella conmutada) para ampliar la capacidad y tamaño de la red.

Las velocidades de transmisión-y los medios soportados han ido evolucionando desde 10Mbps y par trenzado hasta 1Gbps y fibra óptica.

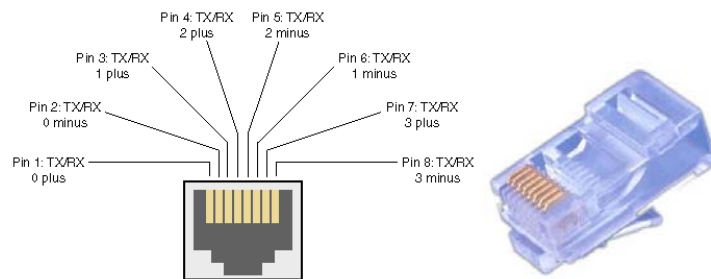


Figura 17 Conector RJ45 [47], [48]

d. USB (1996)

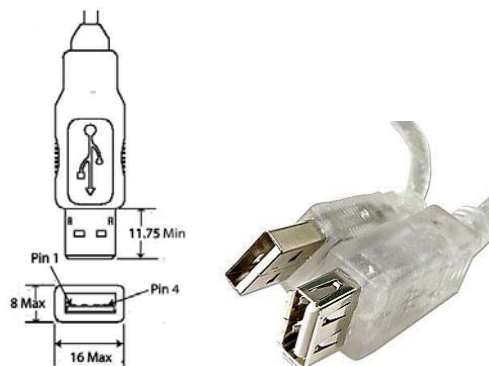
El protocolo USB [6] especifica un bus punto a punto, formando una topología *token ring* liderada por el dispositivo *Host*. El límite de equipos conectables en una misma topología USB es de 127 debido al límite de la etiqueta de identificación de cada equipo. Por lo tanto, se dividirá el ancho de banda entre los dispositivos disponibles.

El equipo *host* puede alimentar al equipo conectado, por lo que es ampliamente empleado en equipos de escaso consumo que no incluyen alimentación propia.

Actualmente el estándar USB ha superado a RS-232 debido a las velocidades obtenidas y mejoras introducidas. Las especificaciones más actuales son:

- USB 1.0 (1,5Mbps)
- USB 1.1 (12Mbps)
- USB 2.0 (480Mbps)
- USB 3.0 (4.8Gbps)

Los cables empleados son par trenzado hasta una longitud máxima de 5m. Se utilizan multitud de conectores, dependiendo del extremo, pero el conector estándar suele ser el tipo A.



**Figura 18 Conector USB Tipo A [49], [50]**

## II. Buses de comunicación modulares

Las 2 arquitecturas de instrumentación más comunes actualmente, definen una estructura más compleja y actual de conexión y configuración de equipo. En este caso, los equipos están más orientados a la comunicación y uso mediante el bus de comunicaciones, evitando interfaces de usuario en el propio equipo, permitiendo una configuración más abierta.

Se incluyen aquí las ya estudiadas VME y PXI, y otras con menos uso en instrumentación industrial, como son PCI, PCIe o compact PCI.

a. VXI

VME eXtension for Instrumentation (VXI) [7], [8] nace en 1987 por un consorcio de empresas (HP, Ramal Data, Tektronix y CDS) a partir de la extensión y particularización del estándar VME e incorporando las ventajas del bus de comunicaciones GPIB.

Aprovechando la gran difusión de GPIB y la estructura modular de VME se añaden características específicas para orientar la estructura a la instrumentación y tratar de concretar un estándar más específico y completo que VME:

- Dispositivos basados en mensajes y registros.
- Mejoras en las capacidades de disparo mediante TTL y ECL.
- Intercambio de datos y señales de forma local.
- Mejora en la temporización y sincronización.
- Aumenta el rendimiento de sistema reduciendo el tiempo de prueba.

La arquitectura del sistema VXI se basa en *racks* de 12 módulos. El *slot* 0, estará ocupado por una tarjeta que realizará las funciones de gestión del bus (estrella, localización y reloj), gestión de recursos, temporización... Se especifican las funcionalidades más concretas de los conectores P1 y P2 de VME, y se añade P3. Hay 7 buses para la implementación de las capacidades:

- **Bus de Reloj:** Dos señales de reloj y una de sincronismo.
- **Bus de Disparo:** 8 líneas TTL y 6 ECL de propósito general. Existen varios protocolos para realizar la comunicación.
- **Bus VME:** Empleado en tareas de comunicación.

- **Bus de suma analógica:** Un solo bus accesible por todos los módulos.
- **Bus en estrella:** El *slot 0* actúa como conmutador entre todos los equipos, permitiendo mayor velocidad de comunicación entre módulos.
- **Bus local:** Comunicación de un módulo con su adyacente. Así podemos permitir una conexión serie de dispositivos.
- **Bus de localización:** Permite conocer que dispositivos y dónde están conectados, mediante conexión directa con *slot 0*.

Los diferentes tipos de comunicación previstos entre módulos son mediante mensajes, registros o memoria compartida.

Actualmente el estándar de 1987 se ha extendido hasta llegar a la versión VXI-10.

b. LXI

En 2005 nace el estándar LAN eXtension for Instrumentation (LXI) [9] a partir del consorcio LXI, con la intención de sustituir en el mercado al estándar más extendido, GPIB. El estándar describe la interfaz de control para equipos de prueba basados en Ethernet y acceso *web*.

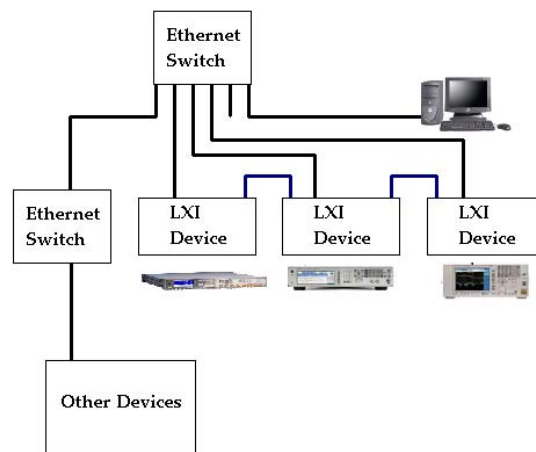
El estándar especifica el uso de los diversos estándares:

- Ethernet.
- TCP/IP.
- Protocolos de descubrimiento (VXI-11, Bonjour...)
- W3C *web browsers standards*.
- Protocolo de temporización de precisión IEEE 1588.
- M-LVDS TIA/EIA-899

Las características principales de LXI son las siguientes:

- Utilización de dispositivos mediante interfaz *web* que sirve cada dispositivo, y posibilidad de interacción dispositivo-dispositivo o controlador-dispositivo.
- Intercambio de señales de temporización y disparo mediante LAN.
- Capacidad de temporización basada en IEEE 1588 mediante la que todos los dispositivos tendrán las mismas referencias temporales para adquirir datos.
- Sistema de cableado de disparo basado en M-LVDS, que reemplaza las redes punto-a-punto de GPIB.

Los dispositivos diseñados para LXI deben tener por lo tanto un servidor *web* a través del que poder manejar todas las opciones de configuración y actuación del dispositivo. Mediante este sistema se pueden utilizar los equipos desde cualquier punto de la red.



**Figura 19 Arquitectura LXI**

Esta tecnología permite multitud de configuraciones. Desde una configuración en chasis, accesible desde red, hasta una configuración completamente distribuida de equipos LXI dispuestos en cualquier punto de la red.



### III. Comparativa de tecnologías

En la siguiente imagen se representa una buena comparación con las diferentes tecnologías estudiadas.

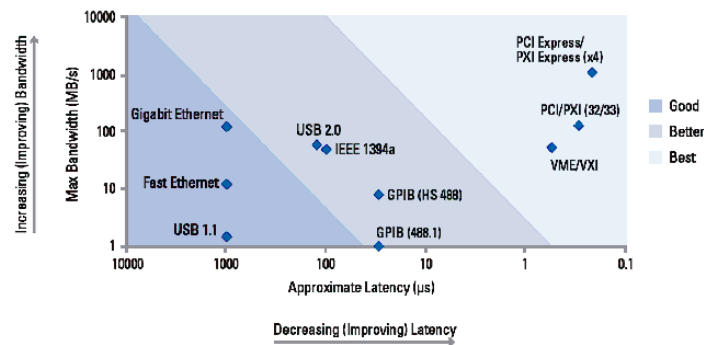


Figura 20 Latencia vs. Ancho de Banda [26]

#### 2.4. Módulo de Alimentación

La alimentación de cada banco será muy particular debido a las enormes diferencias y diferentes necesidades de cada subsistema y cada equipo. Pero habrá, sin embargo, algunas partes comunes a todos, debido a que son propias y comunes en todos los aviones.

Los cableados y colocaciones tampoco tienen un estándar ni unas exigencias más allá de la seguridad. Por lo que en cada banco las distribuciones de cableado, tanto de potencia como de comunicaciones, serán únicos.

##### 2.4.1. Suministro de Potencia AC

Este módulo está conectado a la red eléctrica mediante dos vías, directamente y a través de un **Sistema de Alimentación Ininterrumpida (SAI)**. La función de este módulo es:

- Manejo de *On/Off* del banco manualmente.

- Distribución de la potencia de cada parte del banco, incluyendo potencia 220VAC.
- Protección de todos los sistemas instalados en el banco mediante los correspondientes *circuit breakers* con protección de sobrecarga y los magnetotérmicos. El consumo por lo tanto puede ser controlado manualmente.
- Disponer de un apagado de emergencia para cortar el suministro inmediatamente si se necesita. Se realiza mediante lógica y seta de seguridad.

En la siguiente figura está el frontal del módulo principal de alimentación:

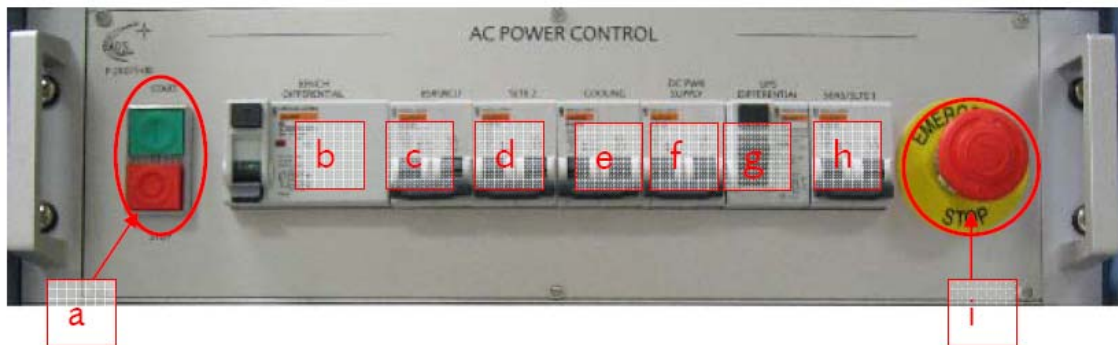


Figura 21 Frontal de alimentación

La funcionalidad de las partes de la imagen es:

- Control general de encendido del banco.** Con este elemento se enciende y apaga el banco.
- Diferencial del Banco.** Elemento de protección cuya función es la desconexión de la instalación eléctrica cuando hay una derivación a tierra mayor de un valor, típicamente 30mA. La configuración normal permite el paso de potencia al resto de elementos.
- Magnetotérmico.** Protección de diferentes elementos en caso de que haya una sobrecarga de corriente o un corte de corriente. Protegerá los elementos que tenga por debajo.
- Magneto.**

- e) **Magneto.**
- f) **Magneto.**
- g) **Diferencial de SAI.** Elemento de protección cuya función es desconectar la instalación eléctrica en caso de derivaciones.
- h) **Magnetotérmico SEAS.** Magnetotérmico de protección de los procesadores del banco.
- i) **Seta de emergencia.** Elemento de seguridad que pulsando corta la alimentación principal del banco. Para volver a restablecer la alimentación, la seta se debe colocar en su posición inicial.

#### **2.4.2. Distribución de Potencia**

Este módulo recibe como entrada la corriente del modulo de suministro de 28VDC y lo divide entre los diferentes equipos del banco, de forma controlada.

Por lo tanto, con este módulo se consigue:

- Alimentar o no alimentar los diferentes equipos que necesitan los 28VDC.
- Los equipos de 28VDC y el cableado está protegido mediante *breakers* instalados en este módulo.

En algunos bancos se dispone de ciertas señales de control de potencia (*PWR Control Signals*) que podrán cortar el suministro de alguna vía concreta. Estas señales vienen del equipo servidor y están activadas por el cliente del subsistema de ensayo. Para desempeñar esta función, todos los interruptores deben estar activos.



**Figura 22 Power Distribution**

En los manuales de cada banco concreto se definirán todas las conexiones correctamente etiquetadas, y se definirán también todas las posibilidades que ofrece el diseño.

### **2.4.3. Suministro de Potencia DC**

Para el suministro de potencia de 28VDC se emplearán, en general, equipos comerciales que transforman 220VAC provenientes del AC Power Control en 28VDC. El equipo suministra 28VDC con un límite máximo de corriente.

Los 28VDC no son un valor aleatorio si no que los equipos de avión, por estándar, en general funcionan con 220VAC o 28VDC. Por lo que siempre tendremos que tener una alimentación alterna y una alimentación de corriente continua. En avión la corriente alterna se obtiene en los motores y filtrando, y posteriormente se consigue la corriente continua. En nuestro caso disponemos de alimentaciones en tierra.



**Figura 23 Fuente 28VDC**

#### **2.4.4. Suministro de Potencia Ininterrumpida**

Para el mantenimiento de la potencia a pesar de cortes momentáneos se empleara una SAI, que mantiene el suministro eléctrico continuo de los equipos conectados a ella. Esto se realiza mediante batería. Cuando se dispone de alimentación, la SAI la deja pasar, en general filtrada, hacia los equipos. Cuando la alimentación se cae, se hace uso de las baterías.

La salida de la SAI está conectada al *AC power control*, por lo que la potencia se distribuye del mismo modo independientemente de que dispongamos de alimentación o no.

En cualquier caso, la alimentación de las baterías tendrá un límite temporal y un límite de potencia que habrá que modelar, dependiendo del número y necesidades de los equipos, antes para decidir sobre el tamaño de la SAI.



Figura 24 SAI MGE

#### **2.4.5. Control Inteligente de Potencia**

Control inteligente de potencia se refiere a una implementación configurable con capacidad para encender y apagar diferentes secciones de alimentación mediante instrucciones o mediante programación previa de las órdenes.

Para esto se hará uso de equipos comerciales de *switching* capaces de funcionar a través de ethernet. La salida del equipo estará conectada al AC *power control* mediante señales de control de potencia ya explicadas.

Un ejemplo es el equipo *Pulizzi IPC3401-NET*.



Figura 25 Pulizzi IPC3401-NET

#### **2.4.6. Tecnologías empleadas en el proceso de optimización**

Se definen las diferentes herramientas necesarias para poder realizar la optimización del sistema de alimentación.

Existen en los bancos de dos herramientas de control de potencia remoto (*Remote Power Control*), una aplicación para realizar la comunicación remotamente con la SAI y los diferentes *scripts* necesarios en los diferentes equipos para realizar el apagado remoto ordenado.

El SW de comunicación con cada equipo particular no siempre se encuentra en el mercado, por lo que será necesario desarrollarlo si no existe.

##### **2.4.6.1. Herramientas de comunicaciones con equipos de control de potencia**

###### **I. Pulizzi**

No existe en el mercado un ejecutable que permita el envío de comandos al equipo *pulizzi*, por lo que se desarrolló en la empresa. Aquí se presenta el uso del ejecutable ya existente en el entorno.

La sintaxis de envío de comandos será la siguiente:

*Pulizzi.exe IP Command [C]*

El parámetro *IP* corresponde a la dirección IP del equipo al que se quiere enviar, no siendo posible especificar el recurso DNS, sólo se podrá realizar con direcciones IP.

Los posibles comandos de envío a través del parámetro *Command* serán los siguientes:

- 0: Activa todas las salidas
- -1: Apaga todas las salidas.
- 1..n: Activa o apaga (dependiendo del estado anterior) la salida número n.

El parámetro opcional *C* se usa para realizar una petición de estado de la salida especificada.

Un código de error se devuelve cuando se ejecuta el comando. Los códigos son los siguientes:

- 5: Salida activada.
- 6: Salida apagada.

## II. BlackBox

Para la interacción con la BlackBox se hará uso del programa *BlackBox\_controller.exe* presentado en el Anexo X.

La sintaxis para el envío será la siguiente:

*Blackbox\_controller.exe HOSTNAME on/off unit\_id*

El parámetro *HOSTNAME* es el nombre DNS el blackbox HW.

El parámetro *on/off* ordenará el encendido o apagado del *Remote Power Control* HW.

El último parámetro *unit\_id* se usa para especificar la unidad HW. Esta unidad se podrá encontrar en la *web* de configuración servida para el HW.

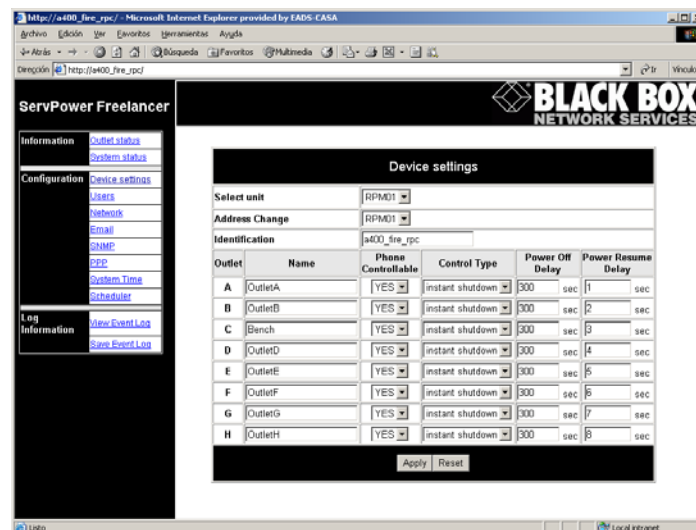


Figura 26 BlackBox Web Configuration

#### 2.4.6.2. Software UPS Merge

En caso de tener UPS de protección, existe el software para realizar un apagado ordenado. Este software es Network Management Proxy (NMP) y Network Shutdown Module (NSM) [51] que se explica a continuación.

La arquitectura elegida para esta instalación es la que se muestra en la Figura. Debido a la configuración del banco, siempre hay un equipo servidor en



el que se puede alojar el NMP, y equipos periféricos en los que instalar NSM, en tantos como se necesite.

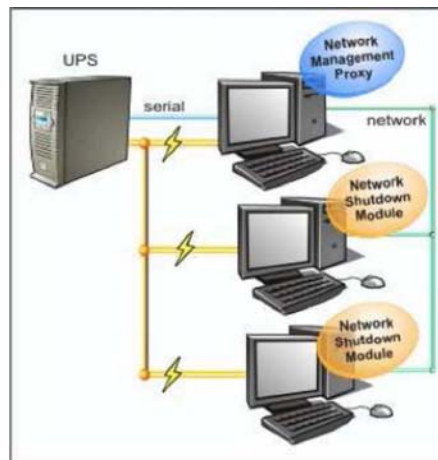


Figura 27 Arquitectura de apagado [51]

#### I. Network Management Proxy

Según la terminología propia de los manuales Merge, el equipo en el que se instalará el NMP será el equipo *proxy* de la SAI.

Este es el software que se necesita tener instalado en aquellos equipos que se pretenda emplear como equipos *proxy* de alguna SAI. Se puede utilizar como *proxy* de una SAI aquel equipo que se desee, tan sólo será necesario tener disponibles puertos USB e instalar el software.

Antes de instalar el SW NMP hay que realizar los siguientes pasos.

- Apagar los equipos protegidos por la SAI.
- Conexión de SAI a la corriente.
- Conexión de cada equipo que debe ser protegido a la SAI.
- Conexión del equipo *proxy* con el cable USB al Puerto USB de la SAI.
- Encendido de SAI y después de los equipos.
- Instalación del software en el equipo que actuará de *proxy*.

Este SW permite gestionar todas las capacidades que ofrece la SAI.

#### **2.4.6.3. RSH**

El equipo cliente debe tener el servidor RSH instalado e iniciado para permitir el apagado remoto del equipo. Algún SO permite el apagado sin necesidad de autenticación, pero al existir diversos SO es mejor estandarizar la estructura independientemente de éste.

Una vez instalado el servidor RSH en las máquinas que no lo tengan, se podrá ejecutar un *script* de apagado remoto. En el sistema, en general, habrá dos tipos de equipos para apagar:

- a. Cliente y servidor con servidor RSH cuya directiva de apagado será:
- b. Equipos de misión con mayor seguridad.
- c. En caso de tener un cliente sobre el que no podamos instalar RSH podremos realizar el apagado con la directiva:

Para controlar la seguridad con el uso de RSH hay que tener el fichero será necesario configurar un fichero de acceso *“.rhosts”*, que especifique los usuarios válidos,

#### **2.4.6.4. Shutdown Script**

Tanto el apagado por debajo de la SAI, como el apagado hacia el *Remote Power Control* se realiza mediante *scripts*. En cada arquitectura propia del banco el *script* será distinto.

### **Capítulo 3. Particularización e Instanciación para una arquitectura genérica de banco de ensayo**

#### **3.1. Alcance**

##### **3.1.1. Introducción**

El objetivo de esta definición, es la especificación de la arquitectura del banco de integración del sistema genérico. El documento resultante de esta especificación debería ser actualizado periódicamente para asegurarse que el documento se refiere a la situación real del banco.

Los objetivos del banco de integración del sistema genérico son:

- Realización de la integración HW/SW y actividades de pruebas del sistema genérico.
- Prueba del sistema interfaz de interconexión con otros sistemas de avión.
- Descubrimiento y aislamiento de errores en los estándares especificados para cada parte del sistema.

##### **3.1.2. Evolución de la descripción**

La presente descripción de arquitectura de banco analizará las siguientes áreas.

- Necesidades y capacidades del banco.
- Diseño preliminar del banco.
- Implementación del banco.
- Validación del banco.
- Actualizaciones del banco.

### **3.2. Requerimientos del Banco**

Los primeros datos necesarios para desarrollar un banco, comprenden tanto las necesidades HW como SW.

Los requerimientos HW se definen tanto para equipos reales como para equipos simulados. Para los equipos reales se incluirán las dimensiones físicas, la alimentación, la refrigeración y los detalles de conexión. Los requerimientos HW incorporan el diseño eléctrico, incluyendo las conexiones externas a otros bancos y las señales necesarias para activar y monitorizar. Las necesidades HW también definen los equipamientos de cada cabina y de paneles. Tras estas especificaciones, será revisado con el cliente, dando como resultado un rediseño preliminar.

Los requerimientos SW incluyen los ICDs, detallando los mensajes, rangos y escalas. Además se incorporan los escenarios y gráficos, incluyendo la capacidad de registro y el modelo de equipamiento lógico de activación. Además, los requerimientos SW son también revisados con el cliente, dando lugar a un rediseño preliminar.

#### **3.2.1. Requerimientos HW del Banco**

En nuestro caso, debido a la privacidad que hay que respetar no se puede definir una arquitectura concreta de banco, por lo que definir unos requerimientos HW [52] concretos que conlleven una arquitectura concreta de banco es imposible. En cualquier caso, sí que podemos definir un pequeño guión de lo que debería contener un documento de requerimientos HW.

No será necesario ceñirse a esta guía HW/SW, si no que se especifican los puntos importantes que debe tener cualquier petición, y posteriormente la estructuración que cada uno haga, y la amplitud de las explicaciones podrá ser mayor o menor dependiendo del usuario.

### **3.2.1.1. Descripción del entorno del banco**

Se deben especificar los bancos o *racks* con los que tendrá relación este nuevo banco, para poder estudiar las posibilidades de reutilización de partes SW y HW.

Las especificaciones correspondientes a entorno de trabajo, estableciendo las condiciones de presión, temperatura, humedad... que debe soportar el diseño.

Aquí se establecerá cualquier particularidad o detalle concreto que sea necesario tener en cuenta para este banco, y no sea algo común en el resto de realizaciones, como pueden ser equipos concretos reales que se exigen, necesidades de seguridad, necesidades temporales...

### **3.2.1.2. Diseño – Desarrollo HW**

Especificaciones concretas de modularidad, y separación de subsistemas y equipos requerida. Se debe concretar si es necesario establecer 2 subsistemas en 2 *sub-racks* completamente independientes, o si por el contrario deben estar necesariamente juntos en un mismo *sub-rack*.

Se especifican equipos concretos que deberán estar en el banco, y que equipos concretos deben estar en cada *sub-rack* concreto. Además irá acompañado de un esquema detallado del conexionado y relaciones entre los diferentes módulos y equipos.

Se explicará de forma detallada los subsistemas y equipos necesarios de todo el banco, así cómo sus interfaces y conexiones.

### **3.2.1.3. Interfaces Humano-Máquina**

Aquí se obtiene una explicación detallada de las necesidades de manejo que requieren las personas que vayan a trabajar con el banco. Entre esto

encajan las definiciones de seguridad, las distribuciones de equipos, equipos necesarios para conexión al banco, estaciones tácticas independientes, y toda la definición de paneles de control y equipos de visualización y monitorización.

En cuanto a la descripción de paneles, será necesario conocer todo aquello que el usuario desea poder manejar desde su posición, y todo aquello que desea ver, ya sea en valores reales o discretos.

Se refleja todo aquello relativo a necesidades humanas, cómo tiempos de arranque máximos, separaciones de *sub-racks*, permisos, conexiones...

#### **3.2.1.4. Necesidades de Sincronización**

Es necesario, por supuesto, definir las necesidades del banco, en cuanto a posibilidades de conexión con otros bancos. Si el subsistema forma parte, como subsistema, de otro sistema más grande, es necesario conocer los deseos de reutilización de éste como parte real de otro sistema mayor, o en caso de no tener una conciencia clara sobre esto, proveer la capacidad de ampliación para futuras revisiones.

También puede existir la posibilidad de que el subsistema no vaya a formar parte de otro sistema, pero en cualquier caso, siempre es preciso otorgar la capacidad de ampliación ante la necesidad de añadir nuevos equipos en la medida que sea posible.

#### **3.2.1.5. Equipos Reales**

En conjunto con el “Diseño – Desarrollo HW”, será necesario especificar concretamente que equipos y subsistemas deben estar representados de forma real en el banco. Y de éstos, es necesario conocer sus conexiones, sus capacidades, sus consumos y todas las especificaciones de dimensiones y características particulares. Multitud de equipos se emplean asiduamente en varios bancos, por lo que se conocen las características de varios equipos

concretos, pero siempre es necesario conocer lo que pide el usuario para conocer sus prioridades.

También será necesario definir los buses de comunicación necesarios para cada equipo, y sus relaciones a través de estos buses, con otros equipos del sistema. Esta definición permitirá disponer de las configuraciones finales y tener la definición completa de la parte HW del banco de ensayo. Los buses de comunicación de aviónica que se emplearán permitirán encontrar equipos de análisis y tarjetas de comunicaciones necesarias para realizar el análisis de las señales y equipos.

### **3.2.2. Requerimientos SW del Banco**

En cuanto a los requerimientos SW [53], al igual que antes no se puede hacer una especificación concreta, pero si una guía sobre aquello que sería imprescindible especificar.

#### **3.2.2.1. Equipos Simulados**

Dentro de las necesidades del banco, hay que especificar aquellos equipos necesarios dentro del banco, pero no son objeto de prueba del sistema. Estos equipos o sistemas serán los equipos simulados. El entorno del subsistema de pruebas no siempre puede obtenerse de manera real, por lo que todas las interacciones de nuestro banco con otros subsistemas deberán realizarse mediante simulación.

Al igual que en caso de sistemas reales, habrá que definir condiciones de relaciones con otros equipos, velocidades y todo aquello que se considere necesario. También se deben definir los buses de comunicaciones que emplearán. Aunque los equipos simulados se vayan a modelar de forma teórica es necesario conocerlos tan a fondo como un equipo real, para integrar estos equipos simulados de modo que no se diferencie la parte simulada de la parte real y el banco sea lo más representativo posible de la realidad.

### **3.2.2.2.      Señales necesarias en el Banco**

A través de los ICD's de avión se definen claramente todas las señales necesarias para el banco. Pero según las necesidades de cada usuario, se pueden añadir otras señales de utilización (como señal de apagado) o simulación, así como algunas señales concretas que sean necesarias para manejar alguna parte del banco, y no existan en los sistemas reales del avión.

El ICD de avión contiene todas las señales que viajan a través de todos los buses de comunicaciones del avión (ARINC429, AFDX, MIL1553...) desde y hacia cada equipo del subsistema. Se actualiza periódicamente ya que los aviones están en continuo progreso, y aunque no suelen variar las señales de los equipos si se pueden añadir nuevos equipos, modificar equipos o añadir nuevos actuadores que representen nuevas señales.

Un avión puede constar de hasta millones de señales, e incluso un subsistema puede llegar hasta 500.000 señales. Con estas cifras es importante disponer de un sistema suficientemente mecanizado de conversión de un ICD de avión a los ficheros de definición de señales del subsistema de ensayo. En este punto, la interacción con el cliente para comprender el correcto alcance de las señales, es fundamental, ya que de no definirse correctamente las señales, el funcionamiento del sistema no será, ni mucho menos, el esperado.

### **3.2.2.3.      Señales monitorizadas**

No todas las señales deben ser monitorizadas, pero en general, si todas las señales pueden ser monitorizadas, ya que existe un subsistema de ensayo con capacidad de visualización de todas las señales definidas. Por señales monitorizadas, se refiere sobre todo a señales imprescindiblemente visibles, y señales que el usuario querrá tener disponibles a golpe de vistazo, a través de *leds*, *lcd's* u otros equipos de monitorización.

Estas señales serán las que se presentarán en la cara delantera del banco mediante representaciones visuales, pero el cliente también puede exigir



paneles definidos en el subsistema de ensayo. Si no se dispone de un equipo de visualización propio del banco, pero si se necesita para realizar las pruebas, los clientes, suelen exigir la representación visual exacta del equipo en subsistema de ensayo.

#### **3.2.2.4. Simulaciones**

Una de las exigencias más importantes del banco son las simulaciones requeridas para el banco. No disponer de equipos reales obliga a simularlos, pero no es la única simulación exigida.

Para poder realizar las pruebas los usuarios definirán los escenarios necesarios. Estos escenarios serán la representación visual de las capacidades del banco. Las simulaciones SW recibirán los datos de los equipos y actuarán sobre las señales necesarias para realizar las pruebas dinámicas si fuera necesario.

Otras simulaciones pueden ser simplemente de representación, como puede ser un radar. Aunque lo que se suele exigir es la representación real que se tendría en el avión tanto de visualización como de actuaciones.

### **3.3. Capacidades del Banco**

#### **3.3.1. Arquitectura Funcional del Banco**

La arquitectura funcional del banco está detallada para conseguir los requerimientos HW del banco y en el diagrama de bloques del banco, los diagramas de flujo de señales desde y hacia los equipos SEAS, una lista de equipos reales (de avión y de laboratorio) y equipos simulados, y las capacidades de conexión.

#### **3.3.2. Diagrama de bloques del Banco**

Esta será la arquitectura del banco de integración del sistema genérico. Las diversas partes estarán comunicadas entre sí o a través de SEAS por los buses de comunicaciones diferentes de los que se dispone. En este caso sólo se dispone de buses MIL-STD-1553, Arinc429, AFDX y señales discretas y analógicas, pero ya sabemos que se pueden gestionar todos aquellos buses que se necesiten.

Cada subsistema del sistema completo puede ser modelado como una parte independiente que dispondrá de unas capacidades de conexión y comunicación con otros módulos. En este caso concreto podrían existir ya unos bancos de pruebas de cada subsistema, pero la integración del sistema completo requiere de la creación de un banco de integración.

En la arquitectura hay de 4 subsistemas de avión y un sistema de control de estos subsistemas. Este sistema de control organizará las comunicaciones y las actuaciones sobre los subsistemas.

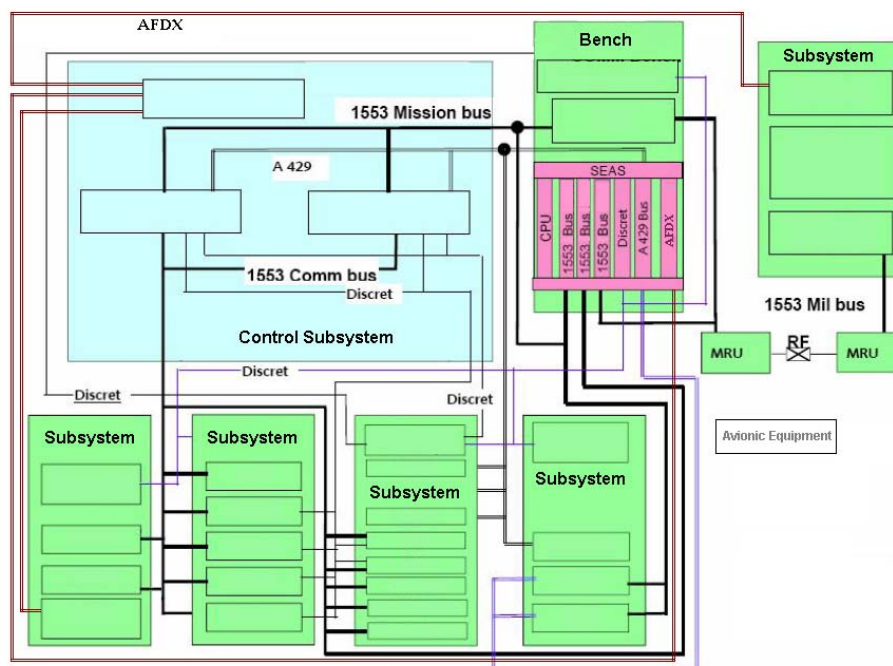


Figura 28 Arquitectura del Banco

Cada uno de los subsistemas, el subsistema de control y el subsistema de ensayo de banco estarán en un *sub-rack* independiente, añadiendo una

modularidad que otorga capacidad para reutilizar, añadir o quitar partes del banco.

### 3.3.3. Escenarios operativos del Banco

Cada escenario será una definición y especificación de equipos de avión reales, otros equipos reales y equipos simulados que hay que configurar para realizar las pruebas concretas.

Esta estructura dispone de dos escenarios posibles.

#### 3.3.3.1. Escenario 1

En el primer escenario, el subsistema de ensayo no simula ningún componente del sistema completo. Por lo tanto, todo está guiado y estructurado con equipos reales de avión.

En este caso el subsistema de ensayo se encarga de la gestión de señales, simulaciones y *recording* y visualización de los datos que se precisen.

Aircraft Real Equipment	Other Real Equipment	Simulated Equipment
<i>Subsystem 1</i> <i>Subsystem 2</i> <i>Subsystem 3</i> <i>Subsystem 4</i> <i>Control Subsystem</i>	<i>None</i>	<i>None</i>

Tabla 1 Escenario 1

#### 3.3.3.2. Escenario 2

En el segundo escenario se simula el subsistema de control. Con esto se consiguen realizar las pruebas necesarias únicamente sobre los

subsistemas periféricos, sabiendo que el sistema de control en este caso es teórico y por lo tanto ajustado a las especificaciones.

Ahora el subsistema de ensayo, además de gestión de señales y simulaciones, se encarga de la simulación de control.

<b>Aircraft Real Equipment</b>	<b>Other Real Equipment</b>	<b>Simulated Equipment</b>
<i>Subsystem 1</i> <i>Subsystem 2</i> <i>Subsystem 3</i> <i>Subsystem 4</i>	<i>None</i>	<i>Control Subsystem</i>

Tabla 2 Escenario 2

### 3.3.3.3. Capacidad de conexión del Banco

Hay algunos bancos que necesitan interconexión con otros bancos, o que permiten una interconexión multisistema para organizar una arquitectura más completa de pruebas, mientras siguen manteniendo una capacidad aislada de pruebas.

En este caso, al no tener una configuración concreta de múltiples bancos no es necesario especificar las posibilidades de interconexión.

## 3.4. Diseño Preliminar del Banco

### 3.4.1. Descripción física del Banco

El diseño preliminar del banco incluye los diagramas de conexiones eléctricas y los planos de diseño de cada cabina. Estos planos incluyen el equipamiento, paneles, refrigeración, modulo de alimentación y detalles de las tarjetas de adaptación. Esta tarea será responsabilidad de una empresa subcontratada que facilitará la documentación relativa.

### 3.4.2. Descripción del Sistema de Estimulación y Adquisición

SEAS aporta capacidad de simulación, estimulación y adquisición. Tiene dos categorías de componentes SW. Una común a todos los testes implementados. Esta primera categoría se configura a través del *Test Bench Definition*. La segunda categoría será particular para cada configuración de test. Esta será la parte definida, ya que será propia para cada banco de pruebas.

#### 3.4.2.1. Arquitectura SW SEAS

Tanto la arquitectura genérica de SW como la arquitectura particular se pueden clasificar en 2 partes:

- Componentes con interacción de usuario, TUI.
- Componentes orientados a señales físicas reales, RTSIG.

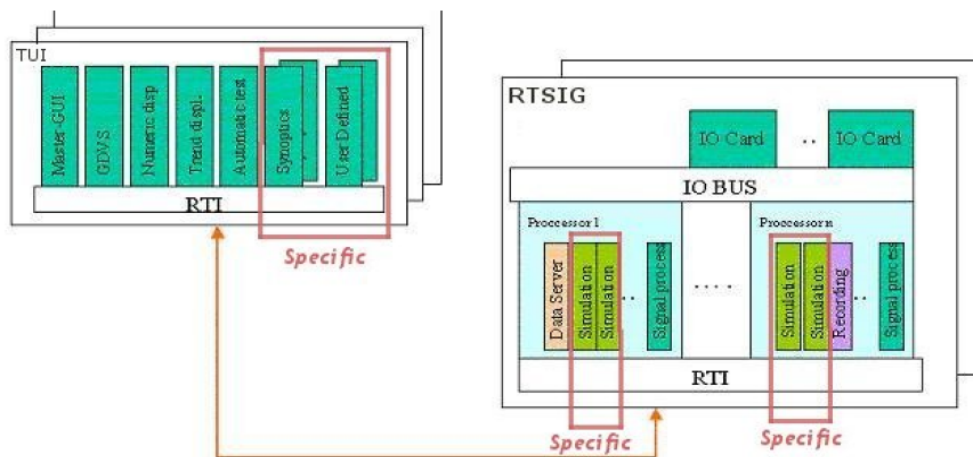


Figura 29 Arquitectura SEAS [19]

Los componentes TUI [18] permiten al usuario configurar los recursos del banco de ensayo de acuerdo con sus necesidades, y permiten también configurar el test, visualizar los valores de las señales en tiempo real, analizar *off-line* los resultados del test y generar reportes de la ejecución. Las características principales de los componentes TUI son, su capacidad de

configuración, permitiendo adaptar todas las herramientas a las necesidades de cualquier sistema de avión.

Hay una instancia TUI para cada usuario conectado al banco. Los componentes principales del SW TUI son:

- **Master-GUI:** Es el acceso del usuario a todas las facilidades del sistema y desde aquí será desde donde se acceda al resto de componentes TUI.
- **General Data Visualization**
- **Numerical Display**
- **Trend Display**
- **Sequencer**
- **Definiciones de Test del usuario**

Los componentes TUI pueden ser divididos en 2 grupos:

**Componentes Off-Line:** Aquí se incluyen aquellos componentes que no necesitan un test en funcionamiento, sólo necesitan la tarjeta procesadora principal y el sistema operativo de RTSIG, y visibilidad de los recursos de test desde el equipo local.

**Componentes On-Line:** Aquellos componentes que usan los servicios de acceso a las señales que ofrece el servidor SW RTSIG y que requieren un test corriendo. Estos componentes necesitan comunicarse con el SW RTSIG. Generalmente se necesita una tasa de refresco menos de 100Hz.

**Real Time SIGnals process (RTSIG)** son componentes del sistema que ofrecen las siguientes funciones:

- Manejo de tarjetas I/O
- Manejo de la base de datos de señales.
- Simulaciones

- *Recording*
- Manejo de definiciones de test, incluyendo *sequencer*.

Los componentes estándar de RTSIG son:

- Controlador RTSIG
- Servidor de Datos
- Grabador de datos
- *Sequencer*

No se define ningún componente específico RTSIG aparte de los componentes estándar.

#### **3.4.2.2. Configuración HW SEAS**

La configuración HW de SEAS involucra 3 partes:

- Instalación física de las tarjetas procesadoras y de adquisición de señales.
- Definición de los componentes SW como simulaciones y gráficos específicos.
- Implementación en el entorno SEAS de la base de datos de señales.

#### **3.4.2.3. Definición física de entorno SEAS**

Los componentes SEAS del banco serán los siguientes:

- 2 Tarjetas procesadoras para simulaciones, *recording* y soporte de interfaces RS-232 y RS-422 (SBS\_VP9)
- 1 Tarjeta de memoria compartida (RAMIX)
- Tarjetas 1553 Dual Channel (AVI-1553-2)
- Tarjetas Arinc429 (SBS-A429-V2)

- Tarjetas AFDX (AVC-FDX-2/4/B)
- Tarjetas Discretas (DSIO-00)
- Placas de salidas Discretas (CM-DO-40)

Aunque son valores variables, en general siempre se dispone de 2 Procesadoras y 1 Memoria compartida. Para cada banco en particular se puede disponer de tarjetas CAN, AFDX, RS422 o RS485, etc.

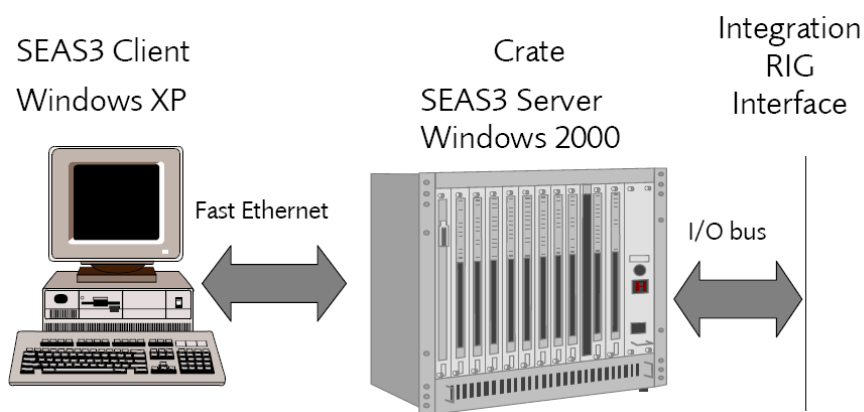
La arquitectura del sistema estará basada en un chasis VME, con bus VME64 y 21 *slots* para tarjetas, diseñado específicamente por la empresa.

QTY	Parte	Descripción
1	20836-220 1	VME64 Chassis, 21 slots
2	VP9	CPU Pentium, VP9-VME64-SBS
1	RM139	8Mb Memory Board Ramix
X	AVI1553-2	MIL-STD1553 I/F
X	API 429	Arinc429 Bus
X	AVC-FDX-2/4/B	AFDX Bus
X	BM-4F001	BCU-DSIO-OBRM
X	CM-DO-40	Discrete Output Board
X	UT-400XX	Tarjetas extensoras PCB de adaptación

**Tabla 3 Descripción de partes**

La arquitectura HW de alto nivel se representa en un dibujo de alto nivel en la siguiente imagen:





**Figura 30 Diagrama de Arquitectura HW [19]**

La interconexión física de las diferentes tarjetas en el chasis se representa en un esquema como el siguiente:

Nº Slot	Tarjeta
SLOT 1	VP9-VME 64 #1
SLOT 2	
SLOT 3	Free
SLOT 4	VP9-VME64 #2
SLOT 5	
SLOT 6	RM139 RAMIX
SLOT 7	AVI 1553 BUS
SLOT 8	AVI 1553 BUS
SLOT 9	AVI 1553 BUS
SLOT 10	ARINC429
SLOT 11	Free
SLOT 12	AFDX BUS
SLOT 13	DSIO #1
SLOT 14	Free
SLOT 15	CM-DO-40 #1
SLOT 16	CM-D0-40 #2
SLOT 17	DSIO #2
SLOT 18-21	Free

**Tabla 4 Distribución de tarjetas**

### **3.4.3. Base de datos de señales del Sistema**

La base de datos de señales del banco se obtiene a partir del ICD de avión correctamente integrado, especificando la velocidad de procesamiento, equipos de origen y destino...

Las señales del RIG son obtenidas a partir del documento de requerimientos, que exigirá unas características de banco. Estas señales serán en general señales discretas de manejo de ciertas partes del banco, tales como pueden ser, apagado o encendido de módulos, desconexión de partes, señales o de control...

### **3.4.4. Simulaciones**

En el escenario 2 definido anteriormente se exige la simulación del subsistema de control.

Este subsistema debe estar corriendo en caso de querer emplear únicamente el escenario 2. En caso de querer usar el escenario 1, será imprescindible que no esté activada la simulación, para evitar errores.

La simulación no es el objeto del proyecto, y ya que el desarrollo es bastante extenso, no se va a presentar aquí. Ya se explicó anteriormente que es una simulación y cual debe ser su estructura.

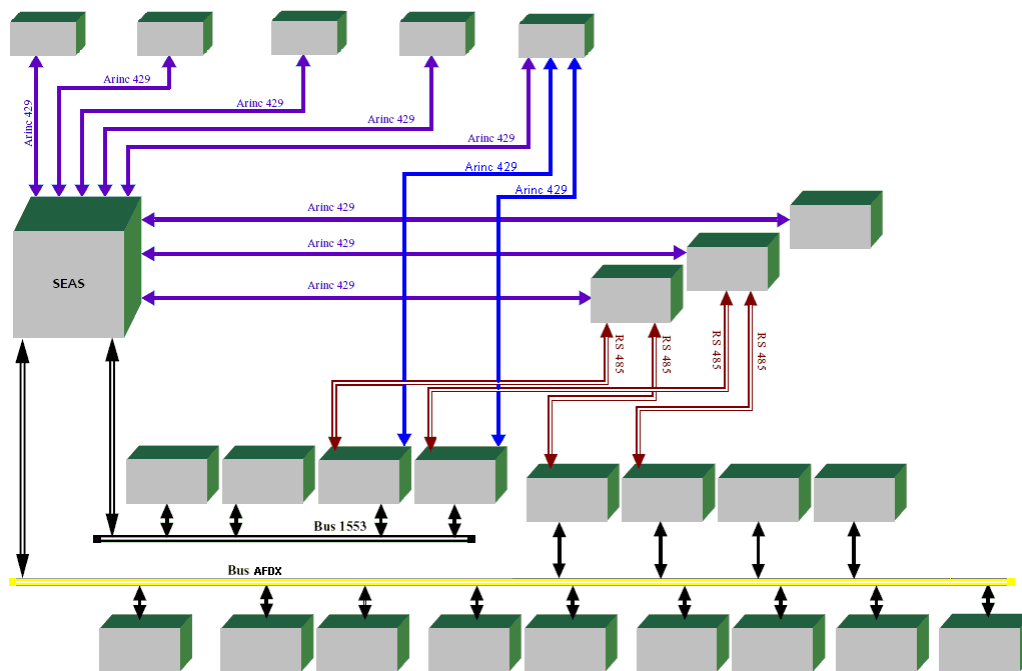
## **3.5. Implementación del Banco**

### **3.5.1. Diseño HW detallado**

El diseño HW detallado es responsabilidad tanto de la empresa como de la empresa subcontratada.

La responsabilidad de la empresa subcontratada corresponde con la realización del diseño mecánico, dibujos, serigrafía, diagramas de cableado, cabinas y conexionado, ensamblado y test de continuidad. Todos estos dibujos y planos estarán a disposición de la empresa EADS.

Para completar el diseño detallado del banco, EADS deberá revisar todos los diseños realizados por la subcontrata.



**Figura 31 Diagrama de bloques del banco**

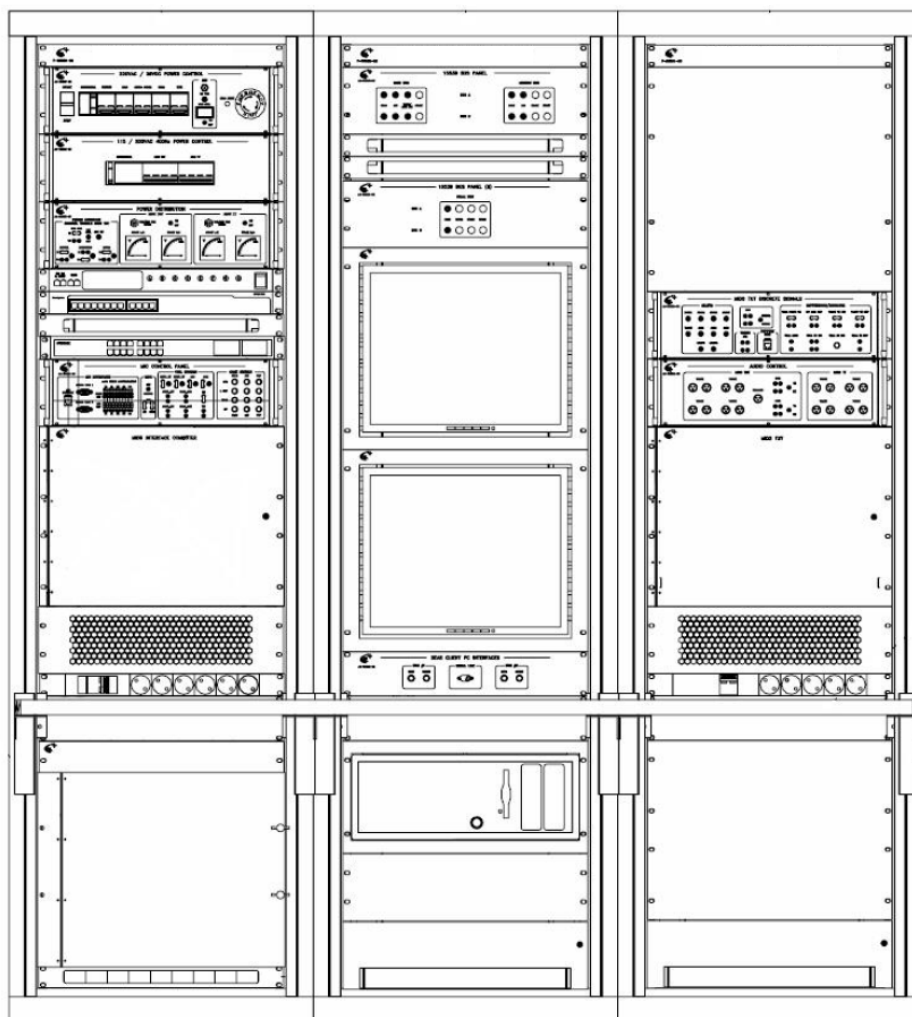


Figura 32 Banco Frontal

### 3.5.2. Electrical Testing

El test eléctrico del banco es responsabilidad de la empresa subcontratada. Debe ser realizado por la subcontrata y personal de EADS podrá supervisar el procedimiento siempre que quiera.

Las pruebas eléctricas que realizaremos sobre el banco se definen a continuación:

- Prueba de continuidad: Comprobación de conexiones de cada uno de los mazos.

- Prueba de continuidad cruzada: Comprobación del aislamiento del pin a prueba respecto al resto de pines del mazo.
- Pruebas de aislamiento respecto a carcasa: Verificación de condiciones de seguridad respecto a equipos y personas.

### **3.5.3. Integración SEAS**

La integración de SEAS en banco es responsabilidad de la empresa subcontratada. Podrá ser supervisado por personal de EADS. Corresponde con las pruebas funcionales del banco.

Las pruebas a realizar son las siguientes:

- Prueba de inspección visual: Comprobación visual de irregularidades en el *subrack*, y comprobación del correcto montaje del esquema.
- Prueba funcional: Validación de requisitos de diseño exigidos para el banco.

### **3.5.4. Configuración SW**

Cualquier banco de integración, sin particularizar los detalles, tendrá esta estructura.

- Cliente del Subsistema de Ensayo
- Chasis (VME, PXI, CPCI...) en el que están las tarjetas de comunicaciones, CPU's...
- Interfaces de equipos de avión con los que comunicarse a través de las tarjetas.

La configuración SW del banco corresponde a la correcta configuración del subsistema de ensayo para la estructura concreta del banco. Esta configuración se realiza mediante la interfaz gráfica del subsistema de ensayo, que desemboca en la creación de los ficheros de configuración.

La estructura de ficheros de configuración del subsistema de ensayo se divide en cinco ficheros:

1. **Hw Config:** Estructura hardware de tarjetas del banco. Se traslada la definición de tarjetas en el chasis hasta un fichero en el que el subsistema de ensayo pueda encontrar la localización, tipo, modo, velocidad... de cada tarjeta.
2. **Signals definition:** Definición de señales de acuerdo al ICD de avión. A partir del ICD se introduce en SEAS la definición de señales necesaria y propia del banco de ensayo empleado.
3. **Scal definition:** Definición de escalados de las señales. Los escalados de las señales son las modificaciones sobre los valores planos de la señal, con la intención de obtener una representación real (unidades, rango, visualización, valores discretos on/off...).
4. **Proc signals:** Definición de simulaciones. Incluyendo cada señal para cada simulación. Cada simulación necesita disponer de un conjunto cerrado de señales. Para poder emplear las señales se enlazarán con las simulaciones mediante un fichero ADA creado con el subsistema de ensayo, donde se define el nombre de señal real en el banco, y el nombre empleado en la simulación.
5. **Bus definition:** Definición de buses de comunicación simulados y monitorizados. Los buses de comunicaciones, así como canales, divisiones dentro de los buses... deben ser especificados al subsistema para poder realizar la correcta asignación de las localizaciones físicas, con la definición de señales disponible en el sistema.

## Capítulo 4. Detección de mejoras y áreas de Optimización

### 4.1. Introducción

Se va a realizar un análisis de los elementos del banco para conseguir comprender y descubrir que posibilidades de mejora existen sobre la estructura de banco actual.

El análisis se va a realizar por módulos:

- Subsistema de Ensayo.
- Módulo de Comunicaciones.
- Módulo de Alimentación.

### 4.2. Subsistema de Ensayo

El subsistema de ensayo empleado en los bancos de integración, se basa en la construcción virtual del banco sobre el que actúa, y cuya configuración estará almacenada en ficheros “.xml”. Actualmente la estructura de ficheros de configuración es elegida por el usuario para cada test de prueba que necesite ejecutar. Sin embargo, la consistencia de estos ficheros a lo largo del tiempo y a lo largo de las diferentes versiones, es inexistente. El objetivo de este apartado es conseguir una **estructura de ficheros y de recuperación de configuraciones dinámica y controlada automáticamente**.

Para la consistencia de ficheros, se emplea generalmente, CVS. El problema que existe es la dificultad de mantener un registro común de las diferentes versiones, y su reposición de una manera sencilla y controlada. CVS garantiza la consistencia, pero tanto la actualización como la reposición se realiza, actualmente, a través del propio interfaz CVS, lo que requiere un mínimo conocimiento del programa. Aunque CVS es integrable mediante funciones en cualquier otro código, esta poco extendido y difundido el uso de este método.

Por sencillo que sea, el usuario no siempre tiene conciencia sobre la importancia de mantener un registro completo de ficheros que poder controlar y seguir la evolución a lo largo del tiempo.

La nueva línea que está siguiendo la evolución de los bancos se basa en la generalización y la capacidad tanto de ampliación como de integración de una manera sencilla. En ese nuevo marco, va a resultar imprescindible tener un buen sistema de control y almacenamiento de ficheros de configuración, ya que las diferentes integraciones y modificaciones sobre el banco repercutirán directamente sobre los ficheros, siendo imposible realizar las modificaciones sobre los ficheros bajo demanda, o mantener la compresión sobre cientos de distintas configuraciones de un mismo banco.

#### **4.3. Modulo de Comunicaciones**

Las comunicaciones de un banco de ensayo entrañan diversos problemas tanto internos al banco, como de integración posterior en subsistemas mayores. El objetivo debe ser **simplificar el entorno de comunicaciones** internas (facilidad de conexión y de comprensión del banco) y externas (facilidad de integración con otros subsistemas y facilidad de introducción de nuevos elementos)

Las comunicaciones internas del banco requieren un estudio muy completo y detallado de todas las posibles configuraciones del banco, y requieren la configuración HW y SW de todas y cada una de ellas. Por lo que un banco, por tamaño, tendrá una capacidad limitada de generalización de configuraciones.

Las comunicaciones externas están ligadas también con lo anterior. Ya que un banco admite un número limitado de configuraciones (por espacio) será necesario crear múltiples bancos para configuraciones que tienen multitud de partes comunes.



Actualmente existen diversas instalaciones con bancos de integración, a distancias considerablemente grandes como para no permitir una conexión de comunicaciones punto a punto entre ellas. Por lo tanto, el uso y configuración de los bancos de integración se realiza exclusivamente en presencia o conexión directa con el banco. Uno de los recursos más importantes en cualquier instalación es el espacio. Un banco de integración SW/HW suele requerir un tamaño de  $5\text{m}^2$ , e incluso dependiendo del banco se puede necesitar hasta  $30\text{m}^2$ . Para mantener un número importante de programas de pruebas se necesitarán instalaciones muy grandes.

Esto tiene solución. Si se dispone de posibilidad de integrar las capacidades de varios bancos en uno sólo, y éste, puede ser accesible desde cualquier punto, tanto para configurar como para utilizar, no sólo se solucionan los problemas de desplazamiento y espacio, si no que además se evita la pérdida irreversible de recursos.

Si un banco queda obsoleto, actualmente no es posible aprovecharlo, ya que cada banco tiene una configuración fija y adecuada para el propósito que se construyó. Sin embargo, generalizando la estructura se pueden reaprovechar las partes de un banco, e incluso actualizarlas sin reconfigurar toda la estructura HW y SW.

Como se explicó anteriormente, los bancos de integración prueban diversos subsistemas que posteriormente se integran en un sistema superior. Pero las pruebas hay que realizarlas sobre los subsistemas individualmente, y posteriormente sobre el sistema completo. ¿Cuál es el problema? Para un sistema con 2 subsistemas, realizaremos 2 bancos para el subsistema y 1 para la integración. Encontrando una correcta solución, se consigue realizar 1 sólo banco que permite ampliaciones, reconfiguraciones y mejoras, incluyendo ambos bancos.

A continuación, se muestra un pequeño análisis del coste para 2 bancos de integración que forman parte de un sistema mayor.

**Tabla 5 Estudio del coste de desarrollo de bancos**

	Banco 1	Banco 2	Banco
Coste de Diseño	15700	15000	20000
Coste de Materiales	34500	38000	35000
Extras	0	0	30000
Total	50200	53000 103200	85000

El coste de cada banco por separado es claramente inferior al coste total estimado de un banco reutilizable. Sin embargo el coste total de ambos bancos será 103.200 €, mayor que el coste del otro banco, cuya funcionalidad es la misma que ambos bancos. Además se deben incluir otras mejoras derivadas como la reducción de espacio, y la capacidad de futuras ampliaciones para su uso como un tercer banco de integración, cuarto...

#### **4.4. Modulo de Alimentación**

El recurso más importante de cualquier instalación es siempre el dinero. Con la configuración actual de los bancos de integración, el encendido y apagado de éstos se hace una tarea compleja que el usuario final del banco no tiene por qué conocer. La intención de la optimización propuesta es **simplificar el encendido y apagado** de bancos para conseguir con un simple clic de ratón o pulsación de botón apagar por completo todas las partes del banco.

Un banco de integración, como se ha visto, consta de múltiples partes, y cada una de estas partes tendrá unas características particulares en cuanto a necesidades de arranque y parada de sus sistemas. El usuario del banco, en general, no tiene por qué conocer todas estas particularidades, por lo que, en general, los clientes de bancos mantendrán encendidos y consumiendo permanentemente los bancos de integración, para evitar pérdidas de información o problemas importantes en la estabilidad de los equipos.

Una necesidad asumida actualmente por todos es la concienciación social en torno al calentamiento global debido al consumo excesivo por parte

del hombre, que provocan una emisión excesiva de gases (especialmente CO<sub>2</sub>) a la atmósfera. Con la solución propuesta se contribuye a minimizar el gasto eléctrico y la contaminación producida.

Se presenta un análisis del consumo del banco para comprender la necesidad de la mejora del sistema de apagado de bancos, así como para concienciar a cada usuario de la necesidad de apagado y arranque de éste sólo cuando sea necesario.

**Tabla 6 Estudio de consumo de potencia de banco**

<b>Sub-System</b>	<b>Consumption</b>	<b>Total</b>
Subsistema 1	28 V	1200 W
Subsistema 2	28 V	300 W
Subsistema 3	28 V	260 W
Subsistema 4	18 V	196 W
Subsistema Control	28 V	2000 W
SEAS	28 V	100 W

Precio del Kw/h = 8,3007 cent €.

Consumo total = 4056 W

La producción de 1 kWh produce, de media en Europa, 0.53 Kg de CO<sub>2</sub>, por lo que cada reducción, por mínima que sea, ayudará mucho al objetivo de reducción de emisiones.

**Tabla 7 Estudio de mejora en la contaminación**

	Horas Mes (h)	Potencia (W)	Consumo (Kwh)	Gasto Mes (€)	CO2 (kg)
Actual	480	4056	1946,88	187,4614151	1031,8464
Posterior	200		811,2	78,10892294	429,936

Precio  
0,083007 €/Kwh

Contaminación CO2  
0,53 kgCO2/Kwh

Por lo tanto, se reducen las emisiones de CO<sub>2</sub> en un 60%.

## **Capítulo 5. Optimización del Banco de Integración**

### **5.1. Introducción**

Tras la detección de áreas de optimización se va a proceder a dar solución a los problemas o posibilidades de mejora detectadas. Al igual que antes, este proceso se va a desarrollar siguiendo los tres módulos existentes en el banco.

- Subsistema de Ensayo.
- Módulo de Comunicaciones.
- Módulo de alimentación.

### **5.2. Subsistema de ensayo: Reorganización de la estructura de ficheros de configuración**

#### **5.2.1. Objetivo**

El nuevo diseño de subsistema de ensayo consigue la integración en éste de un sistema de control de versiones de ficheros para facilitar la tarea de almacenamiento y reposición de las diferentes configuraciones posibles del banco.

#### **5.2.2. Diseño**

##### **5.2.2.1. Necesidades de subsistema de ensayo**

La configuración del subsistema de ensayo se realiza mediante ficheros de configuración “.xml” editables por el usuario en cualquier momento mediante la interfaz gráfica.

La estructura de ficheros de configuración del subsistema de ensayo se divide en cinco ficheros:

- **Hw Config:** Estructura hardware de tarjetas del banco.

- **Signals definition:** Definición de señales de acuerdo al ICD de avión.
- **Scal definition:** Definición de escalados de las señales.
- **Proc signals:** Definición de simulaciones. Incluyendo cada señal para cada simulación.
- **Bus definition:** Definición de buses de comunicación simulados y monitorizados.

El fichero de configuración de la estructura HW, es el que marca el inicio de la configuración de banco, y repercute en toda la estructura posterior. Cuando se conecta el banco, esta estructura ya estará fijada, por lo que si es necesario modificarla, hay que reiniciar el servicio del servidor SEAS en el banco.

El resto de ficheros configuran la estructura junto con el *HW Config* y manteniendo una consistencia entre ellos. Las señales deben mantener un escalado correcto, el *Proc signals* debe incluir las señales adecuadas, *Bus definition* debe estar en consonancia con *HW config*, etc... Por lo tanto, se debe mantener una versión general del estado completo, y no de los ficheros por separado.

Otros ficheros que hay que mantener en consistencia en el proyecto son los ficheros gráficos de representación de señales en paneles, ya que están basados en las señales definidas en el fichero *Signals definition*.

#### 5.2.2.2. **Elección del sistema de gestión de versiones**

La característica de **versionado de proyecto** y no de ficheros, es una característica fundamental, y que caracteriza al sistema **SVN**. Aunque es necesario mantener la versión de ficheros, debido a las necesidades de consistencia e integridad de todos los ficheros conjuntamente, es imprescindible mantener la estructura completa del proyecto.

Otra característica fundamental para elegir SVN por encima de CVS u otros sistemas de mantenimiento de versiones, es la completa definición de API's para integración en programas propios. Tanto CVS como SVN son integrables en otros proyectos, pero por ser más actual y modular, la integración de SVN se hace más sencilla.

Por lo tanto, la elección será SVN. Un sistema más actual y en crecimiento que CVS. Ambos son código abierto, por lo que si fuera necesaria alguna modificación, tendríamos la capacidad de rediseño del sistema.

#### **5.2.2.3. Integración**

La integración de SVN en SEAS requiere de la actuación conjunta de varias personas. Aquí se va a tratar la solución simplificada de integración.

SEAS dispone de interfaz gráfica en cliente que habrá que modificar, servidor SEAS que habrá que modificar y estructura de ficheros que modificar. Se presenta una solución a la nueva estructura de ficheros y aproximando la forma en que se realizará la actualización de ficheros tanto en servidor como en cliente.

##### **I. Estructura de ficheros**

Existen diferentes configuraciones para la ejecución de test en un banco, e incluso cada cliente, puede tener la capacidad de conexión a múltiples bancos, por lo que el usuario no podrá almacenar permanentemente una copia de trabajo para cada configuración de cada banco. Lo que se consigue con SVN es crear una copia de trabajo cuando el usuario lo necesite, y permitirle la opción de mantener esta copia de trabajo en directorio local.

Actualmente se almacenan los ficheros de configuración en un servidor en red, por lo que se sigue manteniendo esta estructura, y se configura el servidor en red como servidor de repositorio de SVN.

El servidor SEAS en el banco, necesita estos ficheros de configuración para comenzar el proceso de pruebas, por lo que debe crearse un directorio de trabajo con la configuración elegida para el test seleccionado. Este directorio de trabajo está también en el servidor de ficheros en red, ya que el banco no almacena estos ficheros localmente.

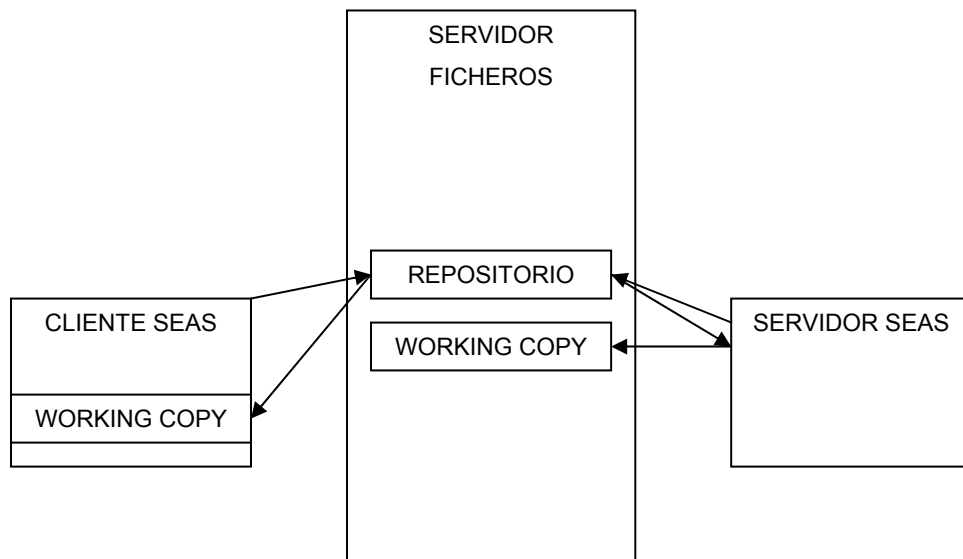


Figura 33 Estructura de ficheros

a. Elección de configuración de banco

Al elegir el banco sobre el que se realizará el test de pruebas, y la configuración correspondiente para éste, se creará la **Working Copy** (WC), o se actualizará en caso de que exista y no esté en la última versión.

Las preguntas al usuario sobre la necesidad de actualizar a la última versión o a la que estime oportuno, o demás consideraciones estarán en la integración final en el subsistema de ensayo. La finalización perfecta del sistema lleva mucho más desarrollo.

El código de integración para realizar el *update* sirve tanto para crear la WC del cliente SEAS, como para realizar la WC del servidor SEAS en el



servidor de ficheros, ya que lo que pasamos como parámetro es un *path* que no tiene porque estar en nuestro disco local.

Para realizar el *update* se necesitará *username* y *password* autorizados, el *path* dónde realizar la WC en disco local y la *url* disponible para el repositorio.

#### b. Modificación y actualización

Cuando el cliente tenga los ficheros en local en la WC, podrá modificar estos ficheros mediante los editores dispuestos en SEAS actualmente. Estas modificaciones no podrán influir en la configuración real hasta que se suben al repositorio, ya que el servidor SEAS los coge directamente del repositorio. Por lo tanto las modificaciones y las comprobaciones de consistencia se pueden realizar en el directorio local, sin influir en el trabajo de banco de otros usuarios.

Como ya se dijo, la actualización se realiza por proyectos y no por ficheros, ya que es imprescindible mantener la consistencia entre los ficheros de configuración. Por lo tanto el código actualizará el directorio de configuración debidamente.

Al igual que para realizar el *update*, se necesitará *username* y *password*, *path* y *url*.

Estas dos capacidades se introducen en SEAS mediante una librería “.dll” con la que se hará uso del *update* y el *commit* cuando sea necesario y con los ficheros que se desee.

#### c. Consistencia y problemas

Los ficheros de configuración deben mantener una consistencia entre todos sus datos. Es necesario mantener esta consistencia, para evitar tener versiones en el repositorio que no puedan ser ejecutadas. Por lo tanto, en un paso previo al *commit*, se realiza el chequeo de consistencia del que ya se dispone en SEAS.

Los conflictos surgidos a partir del *update* o el *commit* de versiones en SVN serán solventados mediante la interacción de los diferentes usuarios, para poder mantener una línea de trabajo correcta y adecuada.

## II. Instalaciones y necesidades

SVN es necesario que esté instalado en el servidor de ficheros que se utilizará como repositorio, así como un servidor Apache 2.x que permita las conexiones exteriores para realizar las diferentes operaciones. Será necesario crear los repositorios adecuados en el servidor de ficheros y ofrecer este a través del servidor Apache 2.x.

Tras la instalación y correcta configuración de los equipos se puede comenzar a utilizar las librerías necesarias para operar con el subsistema de ensayo.

### 5.2.3. Resumen de mejoras

Gracias al sistema de control de versiones integrado en el subsistema de ensayo, se consiguen los siguientes beneficios con respecto al modelo previo.

- I. **Mantenimiento de la línea de trabajo.** Gracias al mantenimiento de las versiones pasadas, no se pierde la línea de trabajo seguida. Manteniendo el control sobre las versiones estables, las modificaciones sobre una configuración no alterarán la versión que ya están terminada y funcionando. No es necesario duplicar configuraciones y mantener mentalmente la estructura correcta de ficheros, esto es precisamente lo que provee un sistema de gestión de versiones.
- II. **Seguridad ante fallos.** Ante cualquier error en la creación de una nueva configuración o modificación de versiones previas, sigue estando presente la configuración correcta disponible. Esto permite la capacidad

de trabajar con mayor tranquilidad y margen de maniobra, permitiendo realizar pruebas y modificaciones sin alterar el correcto funcionamiento.

- III. **Trabajo en red de todo el equipo.** Al disponer del servidor de ficheros, todo el equipo puede trabajar en paralelo en la modificación de la configuración del banco. E incluso se puede utilizar el banco sin que el trabajo que se esté realizando interfiera, ya siempre se puede cargar en el banco la configuración estable aunque se hayan añadido nuevas versiones.
- IV. **Trabajo *off-line*.** Al realizar la WC en disco local, no es necesario estar permanentemente conectados al servidor de ficheros en red. Por lo tanto se pueden crear o modificar configuraciones y validar la consistencia en directorio local, sin ni siquiera interferir o modificar el repositorio de ficheros de SVN.
- V. **Organización sencilla de la estructura de bancos.** Con la nueva estructura de ficheros, se mantienen las configuraciones del banco bajo nombres estandarizados a los que se accede automáticamente. Es imprescindible no interaccionar con el repositorio o con la WC manualmente, para poder mantener la estructura automáticamente desde el subsistema de ensayo. Así se cargan, bajo un nombre identificativo de cada configuración, a ésta de forma completa y asegurando su correcto funcionamiento. Evitando el uso indebido del repositorio se garantiza la consistencia de todas las configuraciones presentes en el banco.

### **5.3. Módulo de Comunicaciones: Inclusión de Matrices de Conmutación y Equipos de Análisis**

#### **5.3.1. Objetivo**

El objetivo del nuevo diseño de inclusión de matrices de conmutación es la modularización y apertura del banco de ensayo. Se obtiene mayor capacidad de expansión, mayor simplificación en la creación de nuevos bancos, facilidad de pruebas sobre el funcionamiento del propio banco y mejora en la integración

de nuevos componentes tanto de medida como partes propias del subsistema en pruebas.

### **5.3.2. Rediseño de la estructura de banco**

#### **5.3.2.1. Elección de la tecnología**

Se debe elegir la nueva tecnología sobre la que basar el nuevo diseño de banco de ensayo. Se podría realizar el diseño basándose en la tecnología empleada actualmente, VME o PXI, pero es preciso realizar un análisis de objetivos para valorar si hay otra opción mejor.

Actualmente todos los equipos de instrumentación necesarios en un banco están montados sobre el mismo chasis (varios chasis en caso de necesitar más *slots*), pero las matrices, permiten utilizar multitud de equipos, independientemente del tipo y de su localización, ya que el uso de las matrices no depende de la instalación de los equipos, sólo se necesita la conexión física de las entradas/salidas a los puntos de la matriz. Por lo tanto, se puede disponer de una configuración **completamente distribuida**, en la que una matriz suficientemente grande, pueda reutilizar diferentes equipos, tan sólo, reconfigurando la conmutación.

Un problema que surge cuando se necesita una configuración distribuida es la **gestión de la temporización y sincronización** entre equipos, a la hora de tomar datos. Es imprescindible tener en cuenta la temporización en una configuración distribuida.

Los bancos de ensayo no emplean equipos de instrumentación con interfaz gráfica para que sean empleados por el usuario, si no que **se utilizan tarjetas de comunicaciones** que se configuran y utilizan mediante el subsistema de ensayo.

Las posibilidades de las que se dispone son las siguiente:

- Las tecnologías GPIB o Ethernet, permiten una configuración distribuida mediante topología estrella o línea. Sin embargo, GPIB está orientada a equipos sobre los que recoger datos, no está orientada a configuraciones modulares de tarjetas. Ethernet puede permitir cualquier comunicación, sin embargo no define todo el entorno necesario para el montaje en chasis.
- VME, PXI o VXI están orientadas a la configuración modular que se persigue, basada en equipos sin interfaz para uso manual (tarjetas) y una amplia definición del entorno necesario para la utilización en instrumentación. Sin embargo, todos se basan en una configuración, aunque pueda ser ampliable, acotada en el chasis que definen. No existe la posibilidad de distribuir todo el sistema de comunicaciones a distancias mayores que el propio chasis.
- LXI también está basado en una configuración modular y también considera el montaje sobre chasis (PXI). Considera un sistema distribuido basado en las comunicaciones de área local con Ethernet orientado a la instrumentación, haciendo uso de protocolos de temporización para mantener la consistencia de los datos a través de la red.

**LXI** será la elección por lo tanto más adecuada y que permite cumplir las exigencias previstas:

- Instrumentación distribuida.
- Comunicación de cualquier equipo con la matriz de conmutación.
- Posibilidad de configuración de equipos y matrices a través Ethernet (red local).
- Consistencia en la temporización de la red.

#### **5.3.2.2. Elección de Matrices**

La inclusión de las matrices en el diseño es algo a priori bastante sencillo, y que aporta importantes mejoras, pero aunque el concepto general es

sencillo, la particularización del diseño en cada caso particular, implica varias fases complicadas.

#### I. Elección de las necesidades de conmutación

Este es el punto más complicado. El objetivo es hacer mucho más amplio y general el concepto de banco de ensayo, sin embargo tampoco se puede permitir perder recursos a costa de particularizar cosas que son generales en sí mismas.

En un banco de ensayo existen comunicaciones entre equipos que serán, independientemente de la configuración de banco, permanentes e inmóviles. Por lo tanto, no hay que introducir comunicaciones fijas dentro de la matriz, ya que esto provocará la ocupación de la capacidad de la matriz en actividades que no necesitan ser reconfiguradas.

En el banco estudiado, existe siempre una comunicación ARINC429 común a los 4 subsistemas, por lo tanto, no es necesario incluir esta línea de comunicación dentro de la matriz, ya que independientemente de los subsistemas que se incorporen, o la configuración que se desee elegir (equipos reales, simulados, nuevos equipos...) estas comunicaciones deben estar así siempre.

Sin embargo, si se necesita capacidad de simulación del subsistema de control, por lo tanto las comunicaciones entre el subsistema de control y los subsistemas 1 a 4, si deben pasar por la matriz de conmutación. En caso de simular el subsistema de control mediante SEAS, se puede conmutar esta simulación a través de la matriz, sin necesidad de cableado adicional.

En cualquier banco en el que se desee incluir las matrices, se debe realizar este mismo análisis:

- Comunicaciones fijas
- Comunicaciones configurables

## II. Elección del tamaño de la matriz

Enlazando con el apartado anterior, el análisis sobre las capacidades de conmutación, permite elegir el tamaño y condiciones de la matriz de conmutación. Pero existirán otros criterios necesarios para decidir cuál es la mejor opción.

- a. Conocer el ancho de banda máximo necesario para todos los canales de comunicación.
- b. De acuerdo con las capacidades de conmutación del apartado anterior, elegir el número de puntos de cruce y la configuración “filas x columnas”.
- c. Dependiendo del tipo de comunicaciones a conmutar, elegir la potencia permitida por la matriz.
- d. Analizando las comunicaciones, decidir si son necesarias capacidades adicionales en la matriz, como suma o resta de señales dentro de las matrices.

Dependiendo de las necesidades y capacidades económicas y de recursos, se pueden elegir matrices que ofrezcan otras capacidades adicionales.

## III. Necesidades de configuración y manejo

Entre los estudios previos que hay que realizar sobre las exigencias en el banco de ensayo están las necesidades de configuración y utilización de los equipos.

La conmutación de los canales de comunicación se puede realizar de dos formas, dinámica o estática. Dinámicamente exigiría la realización o uso de un software con interfaz humano accesible, capaz de otorgar una visibilidad en tiempo real sobre la configuración actual de las matrices, y permitir su

modificación. En el caso estudiado, y en general, no se va a necesitar esta habilidad.

La mayor parte de los bancos permanecen en una configuración estable a lo largo del tiempo. Es decir, no sólo no necesitan configuración dinámica si no que debemos establecer las necesidades para permitir configuración estática.

#### IV. Comparativa

Existen muchos criterios de clasificación, como son los puntos de cruce, o el tipo de conexión (single pole/two pole). Se presenta una comparativa de modelos y precios.

**Tabla 8 Comparativa matrices de conmutación en el mercado**

Modelo	FxC	N <sub>x</sub>	Protocolo	Conmutación	BW (MHz)	Potencia (W)	Config.	Precio(€)
Pickering 60-550 /551	128x8	1024	LXI	Reed	5	60	SP	10828
Pickering 60-510 /001	56x33	1848	LXI	Reed	DC	10	SP	27875
Pickering 60-551 /002	128x4	512	LXI	Reed	5	60	SP	8367
Pickering 60-552 /003	32x64	2048	LXI	Electromecánico	5	60	SP	18583
Pickering 60-554 /003	96x16	1536	LXI	Electromecánico	5	60	SP	15546
Pickering 40-560 /021	92x8	1472	PXI	Reed	40	10	2P	8283
Pickering 40-560 /121	138x16	4416	PXI	Reed	50	20	2P	25737
Pickering 40-560 /121	46x8	736	PXI	Reed	50	20	2P	5310
Pickering 40-560 /121	92x16	2944	PXI	Reed	50	20	2P	17729
Pickering 40-562 /121	99x16	3168	PXI	Reed	50	20	2P	15442
Pickering 40-565 /102	96x8	1536	PXI	Electromecánico	25	60	2P	7781
Pickering 40-563 /221	96x8	1536	PXI	SSR	40	20	2P	6448
NI	64x4	512	PXI	Reed	25	10	2P	4699



Modelo	FxC	N <sub>x</sub>	Protocolo	Conmutación	BW (MHz)	Potencia (W)	Config.	Precio(€)
PXI-2532								
NI PXI-2536	68x8	544	PXI	FET	1	10	SP	2749
NI PXI-2534	32x8	256	PXI	SSR	2	55	SP	3499
NI PXI-1129	32x4	256	SCXI	Electromecánico	10	30	2P	3699

SP ≡ Single Pole

2P ≡ Two Poles

#### a. Puntos de cruce vs. Precio

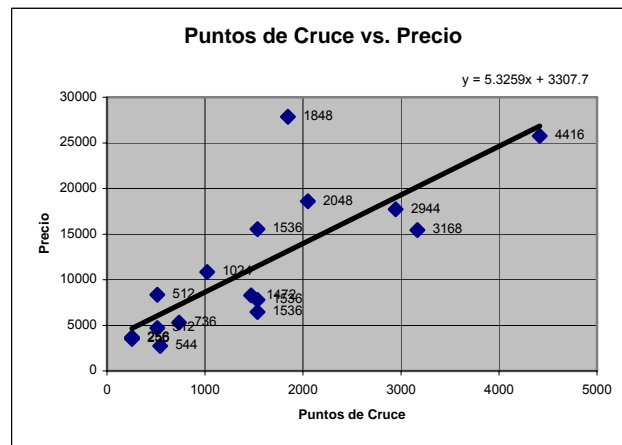


Figura 34 Puntos de Cruce vs. Precio

El precio aumenta con el número de puntos de cruce, pero no hay relación con respecto a la tecnología. Mirando la tabla se observa que las tecnologías mecánicas (Reed o electromecánico) son más caras que las tecnologías sólidas (SSR o FET)

#### 5.3.3. Diseño sobre un caso particular

Con respecto al caso particular analizado en el capítulo anterior, se estudiarán cuáles son las modificaciones y reestructuraciones necesarias para realizar la inclusión de la matriz. Al ser un banco bastante general, por razones de privacidad, la capacidad de particularización para realizar el nuevo diseño

no será completa pero se puede realizar un diseño igualmente genérico y descriptivo de lo que significa la mejora propuesta.

#### 5.3.3.1. Necesidades de conmutación

La valoración de las necesidades de conmutación, es uno de los puntos más complicados en la realización de un nuevo diseño.

En general, el proceso de introducción de nuevas matrices se realiza sobre nuevos diseños, por lo que se parte desde cero para decidir cuáles son las necesidades. En este caso, se parte de un diseño ya realizado, por lo que hay que dividirlo para, en cierto modo, volver a reorganizar la estructura.

Al ser un banco sencillo, sólo se necesita la capacidad de simulación del subsistema de control, por lo que éste subsistema será el que deba estar a través de la matriz de conmutación.

Todos los equipos de instrumentación, tarjetas de comunicaciones... están conectados a la matriz directamente, ya que son los encargados de recoger y enviar datos de equipos. Estos equipos siempre estarán conectados directamente a la matriz ya que ante una ampliación o reconfiguración, siguen siendo los encargados de visualización e interacción en las comunicaciones.

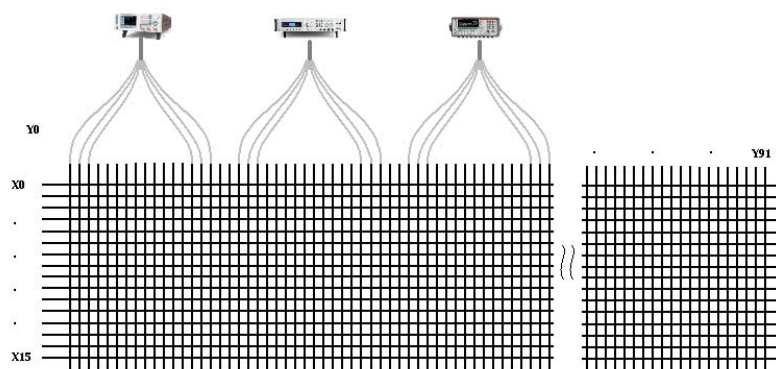


Figura 35 Configuración de matrices

Por lo tanto se necesitan las siguientes conexiones:

- **ARINC429:** 1 par
- **MIL1553:** 3 pines
- **Discretas:** 1 pin

Sin embargo, los buses de comunicaciones de ARINC429 no se emplearán simultáneamente, si no que se empleará uno u otro de los buses. Podemos sacar una parte de las conexiones de ARINC429 de la matriz.

Finalmente, las exigencias de conectividad con la matriz son las siguientes:

- **ARINC429:** 3 x 1 par = 3 pares
- **MIL1553:** 6 x 3 pines = 18 pines
- **Discretas:** 24 pines

Se necesita una matriz con al menos 49 conexiones en columnas. Para encaminar todos estos datos, analizando la máxima utilización, es decir, con todos los equipos reales y todas las tarjetas, se tiene: 24 discretas + 2 pines ARINC429 + 3 pines MIL1553 + 3 pines MIL1553 = 32.

Para permitir un margen se podría emplear una matriz 64x64, o ya que se dispone de capacidad de ampliación siempre que se necesite, se ajusta a lo analizado y se emplea una matriz 64x32.

### 5.3.3.2. Necesidades de configuración

El diseño, así como la mayoría de bancos de ensayo, no necesita la configuración dinámica de las matrices, pero igual de importante es definir las características necesarias para realizar la configuración estática.

- **Configuración:** La configuración inicial de la matriz se debe realizar a través del subsistema de ensayo. No se puede exigir al usuario que

conozca la configuración y la forma en que se debe realizar la configuración de la matriz, por lo que es preciso realizar, de alguna manera sencilla, la interacción entre usuario y matriz.

- **Consistencia:** Una de las capacidades que aporta la matriz de conmutación, junto con la optimización propuesta en el apartado anterior, es la posibilidad de realizar nuevas configuraciones y volver a configuraciones anteriores. Al tener un diseño ya correctamente estructurado, con tan sólo elegir la configuración adecuada, se vuelve a versiones anteriores. En este caso, se puede elegir entre una configuración con equipos simulados o equipos reales, sin necesidad de modificar nada más que la configuración de la matriz.

#### 5.3.3.3. Diseño particular

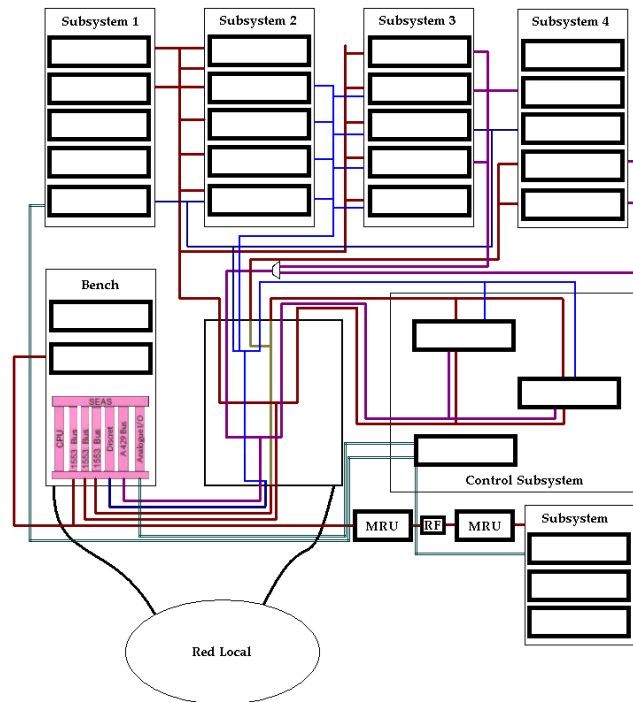
En la particularización del diseño también hay que tener en cuenta los detalles más pequeños. La matriz de conmutación, no es específica de un protocolo, si no que tiene puntos de conexión a los que podremos conectar cualquier tipo de señal dentro de los límites de la matriz. Por lo tanto, hay que convertir el cableado (ARINC429, 1553...) a los conectores específicos para las matrices.



Figura 36 Conexiones de la matriz

Las modificaciones al cableado de los equipos se realizan de manera que mediante un panel de conexiones se pueda conectar a la matriz de conmutación cualquier equipo.

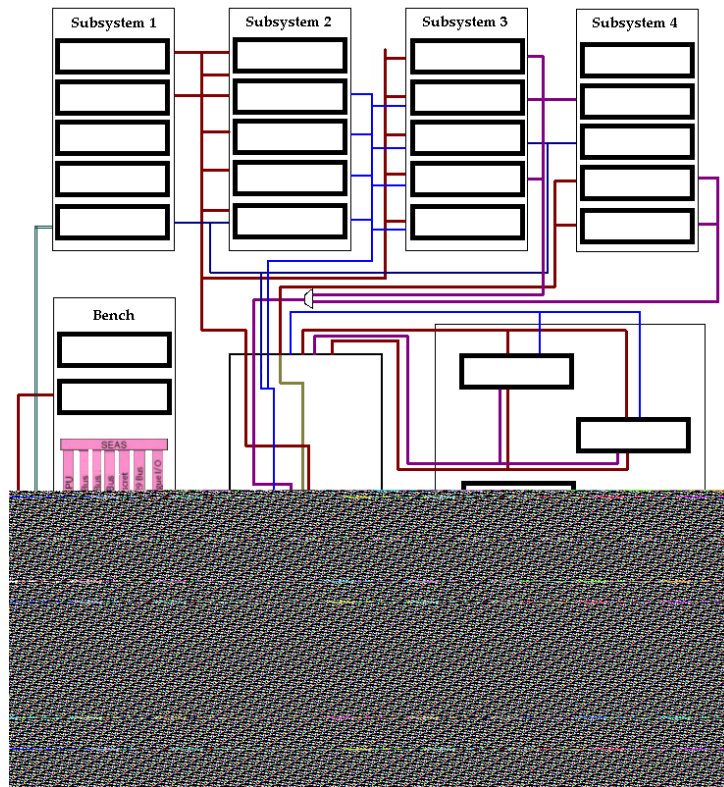




**Figura 38 Diseño con matriz y sistemas reales**

Como se observa, en éste caso, sólo se utilizan las tarjetas de instrumentación para escuchar y analizar cada canal. Hay que introducir señales simuladas que por tener el sistema de avión en tierra, nunca se tendrán de modo real. Con respecto a los sistemas reales, sólo se analizará el funcionamiento y se simulará aquello que falte.

En un diseño con el **subsistema de control simulado** si es necesario simular todas las comunicaciones de éste, además de escuchar a los sistemas reales.



**Figura 39 Diseño con matriz y subsistema simulado**

El diseño se diferencia en la configuración de la matriz. Se reenrutan los distintos canales, y se configura el subsistema de ensayo para que se ajuste al nuevo diseño.

Las diferencias se reducen a dos configuraciones software muy sencillas:

- Reconfigurar la matriz.
- Reconfigurar subsistema de ensayo.

Si se quisiera realizar esta operación sin disponer de la matriz de conmutación, habría que reestructurar el cableado del banco, o prever esta situación en el diseño inicial, y duplicar las comunicaciones para poder operar en ambos casos.

#### **5.3.4. Resumen de mejoras**

Los beneficios obtenidos con el nuevo diseño son los siguientes:

- I. **Capacidad de introducción de nuevos elementos.** Ahora están las comunicaciones centradas en la matriz de conmutación, por lo tanto, cualquier nuevo instrumento o equipo que se deba introducir, sólo será necesario conectarlo a la matriz y reconfigurar filas y columnas para obtener el nuevo funcionamiento.

Un ejemplo, añadiendo un subsistema al diseño anterior, sería el siguiente.

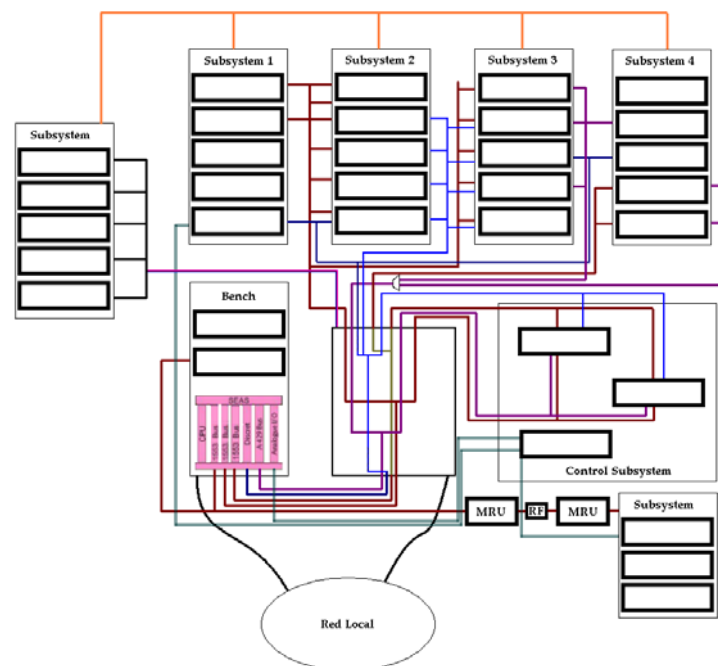


Figura 40 Diseño con nuevo sistema

Para la introducción del subsistema hay cableado específico que se conecta a otros subsistemas, y que no variará. A la matriz se conecta todo aquello susceptible de ser reenrutado. No se ha modificado nada en el chasis, ni se han modificado cableados ya existentes, tan sólo se introducen elementos, otorgando sencillez.

- II. **Facilidad de rediseño de nuevos bancos.** Teniendo las comunicaciones centralizadas se pueden, mediante las mejoras del apartado anterior, y la inclusión de la matriz, emplear los equipos existentes en un banco para su uso en distintos programas de aviónica. No es necesario crear bancos desde el principio. Incluyendo o evitando



algunos elementos, se consiguen nuevas configuraciones exigidas por un usuario.

Por lo tanto, aunque el diseño de un banco con matrices de conmutación, exige más esfuerzo y recursos, se simplifica la construcción de nuevos bancos, minimizando el tiempo global.

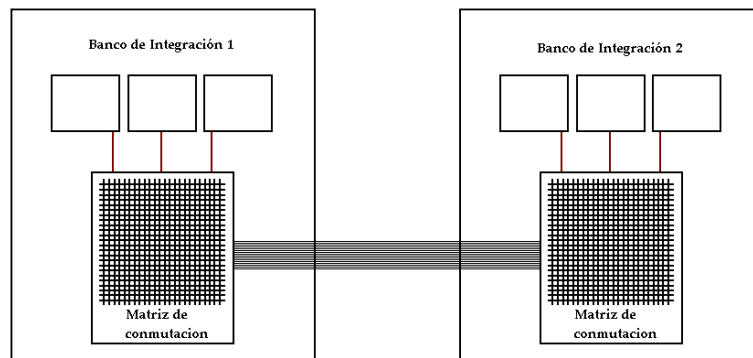


Figura 41 Integración de múltiples subsistemas

Existirán dos subsistemas cuyo funcionamiento está probado por separado, por lo tanto, la integración sólo necesita comunicar las matrices, ya que todo lo demás funciona independientemente. En esta integración sólo se debe modificar una vez más, las configuraciones de las matrices y la configuración del subsistema de ensayo.

**III. Facilidad de testeo de cableado.** Entre las matrices encontradas en el mercado, existe la posibilidad de configuración de la matriz para chequeo. Actualmente una de las partes más costosas en la construcción del banco, son las pruebas de continuidad, aislamiento..., de los bancos. Con la matriz, esto se puede realizar automáticamente, ya que todo el cableado llega a la matriz, y éstas tienen capacidad de medir los parámetros que se le exigen, resistencia, capacitancia, continuidad...

Con esto se ahorran muchos recursos temporales en el proceso de fabricación, ahorrando por lo tanto en recursos económicos.

**IV. Capacidad indefinida de ampliación.** En el primer diseño, se elige una matriz de configuración adecuada para las necesidades actuales, pero si en el futuro fuera necesario ampliar la configuración quizá habría que

ampliar también la matriz. Las matrices elegidas permiten la ampliación tanto en filas como en columnas sin pérdida de consistencia de las señales, por lo que la ampliación del banco se puede realizar cuando se necesite.

- V. **Sistema configurable remotamente.** Al estar basado en el estándar LXI se configura, tanto matrices como instrumentación, remotamente, por lo que no será siempre estrictamente necesario estar en presencia física del banco de integración.

#### **5.4. Módulo de Alimentación: Apagado Remoto y Ordenado de Bancos de Ensayo**

##### **5.4.1. Objetivo**

Hoy en día, el número de bancos activos está en crecimiento, por lo que es necesario un sistema sencillo de apagado de bancos tanto para que el usuario pueda apagarlo remotamente, o de una manera muy sencilla delante del banco.

Aquí se especifican los procesos y facilidades necesarias para conseguir realizar estas tareas automáticamente, y de manera muy sencilla. Además se especifican los equipos HW necesarios para el correcto uso de la aplicación.

La idea original es la de realizar un apagado ordenado y remoto siempre que sea posible, de los bancos con el menor coste económico y físico posible. Esto quiere decir, que hay que evitar siempre que se pueda realizar muchas tareas para el apagado. ¿Que se pretende conseguir? Sencillez en cuanto al apagado remoto y presencial del banco.

Primero hay que mantener un esquema que sea aceptable y homogéneo, y que permita crear una configuración única. La mejor opción es que todos los equipos cuelguen de una SAI y a su vez del *Remote Power Control*, porque esto permite apagarlo todo de una manera muy sencilla. No

siempre se puede conseguir esta premisa, por lo que habrá que barajar las otras opciones.

#### **5.4.2. Arquitecturas de Banco**

Para generalizar la mejora a diversos bancos, no se puede centrar en una jerarquía muy concreta, por lo que es necesario estudiar las diferentes arquitecturas que se pueden encontrar, y realizar el trabajo encaminado a que sea válido para todas ellas.

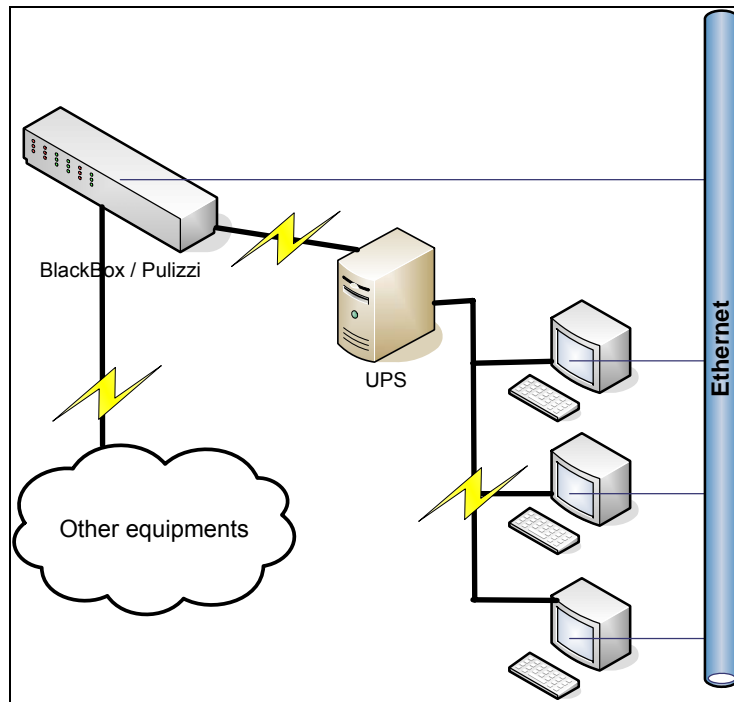
Existen generalmente 4 topologías:

- Banco con SAI y *Remote Power Control*.
- Banco con SAI.
- Banco con *Remote Power Control*.
- Banco sin recursos de apagado adicionales.

##### **5.4.2.1. Banco con SAI y Remote Power Control**

En esta arquitectura hay protección por SAI, y capacidad de apagado remoto. El apagado por lo tanto podrá ser remoto enviando los comandos pertinentes al equipo.

Los equipos no conectados a la SAI serán apagados remotamente, mientras que los equipos protegidos por SAI seguirán el protocolo de apagado.



**Figura 42 Arquitectura SAI y Remote Power Control**

El apagado de equipos se realiza mediante los scripts presentados en el Anexo XIII.

#### **5.4.2.2. Banco con SAI**

En este caso no se puede realizar el apagado remoto debido a que no se dispone de un equipo capaz de gestionar esta tarea. El único modo de realizar el apagado es directamente delante del banco.

Igualmente, si que se dispone de SAI por lo que se puede realizar el apagado ordenado para los equipos por debajo de ésta.

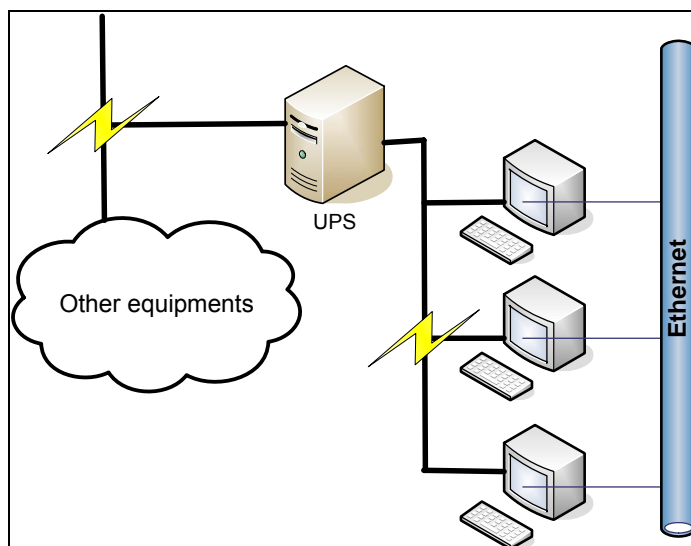


Figura 43 Arquitectura SAI

#### 5.4.2.3. Banco con Remote Power Control

Ahora no hay SAI de protección, pero si existe un equipo de control de apagado remoto.

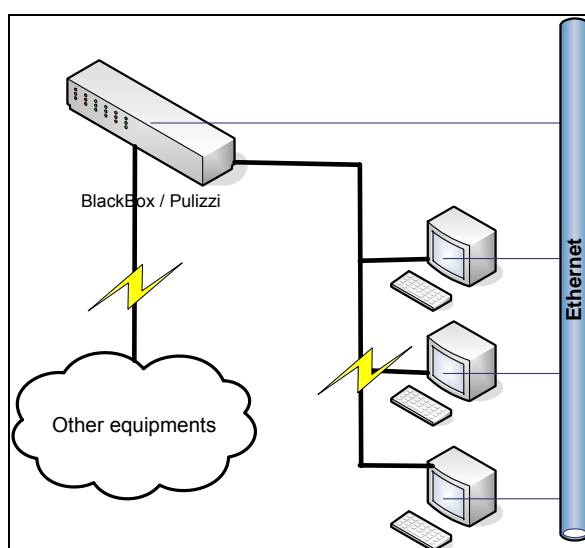


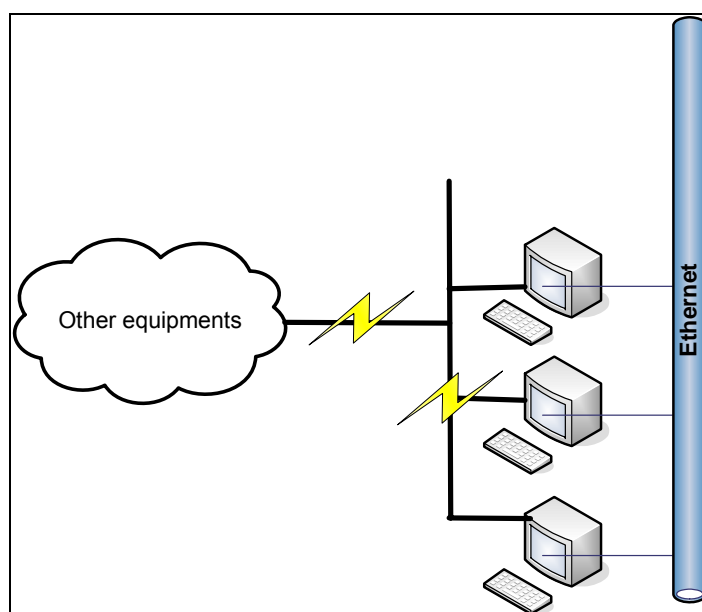
Figura 44 Arquitectura Remote Power Control

Si no se dispone de SAI de protección de los equipos, el apagado de los equipos hay que realizarlo a través del *script*, enviando los comandos apropiados, para posteriormente realizar el apagado del banco completo.

Hay equipos que necesitan de un apagado ordenado antes de quitar potencia, por lo que antes de apagar el banco, hay que apagar correctamente cada equipo especificado.

#### **5.4.2.4. Banco sin recursos de apagado adicionales**

En este caso no hay capacidad de apagado remoto ni protección de equipos mediante SAI, por lo tanto sólo se puede realizar un apagado de los equipos, y posteriormente, realizar manualmente el apagado del banco.



**Figura 45 Arquitectura sin HW adicional**

A pesar de no tener equipos adicionales de protección y apagado remoto, hay que realizar el apagado ordenado de los equipos que lo necesitan, para posteriormente realizar un apagado manual del banco completo.

#### **5.4.3. Apagado ordenado desde SAI**

##### **5.4.3.1. SAI**

La SAI es el elemento de distribución de la corriente del banco. Está provisto de batería para evitar en la medida de lo posible la desconexión bruta de los equipos. También tiene un puerto de comunicaciones USB, a través del

que se puede configurar y a través del que se indicará a nuestro servidor la caída de la corriente, y la necesidad de apagar el sistema.

Las SAI no son configurables de forma aislada, requieren de un contacto permanente con un equipo. Hay varias configuraciones posibles para manejarlas:

- a. Haciendo uso de un *proxy* para una SAI
- b. Mediante tarjeta de comunicaciones.

Para ahorrar lo máximo posible en cada banco, se utiliza la SAI, de la que cuelgan los servidores, como principal, gestionada a través de un *proxy*, y el resto de SAI's se apagan a través del circuito de apagado que proveen éstas (ROO), haciendo uso de lógica adicional.

Las estaciones SAI tienen una lógica adicional en la parte trasera para permitir apagado/encendido mediante electrónica.

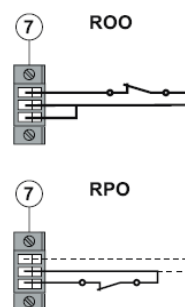


Figura 46 Circuitos de ROO y RPO

- ROO (**R**emote **O**n/**O**ff): es una función de marcha/parada a distancia de ella, que permite recurrir a la acción remota.
- RPO (**R**emote **P**ower **O**ff): es una función de parada a distancia del SAI, que permite dejar sin tensión al conjunto de los equipos conectados a ella mediante un contacto remoto del usuario. El re arranque sin embargo requiere acción manual.

La mejor opción es utilizar ROO, que permite tanto apagar como encender de forma remota.

Esto se emplea para apagar las SAI no conectadas directamente al servidor, por lo que es necesaria una señal que esté activa cuando el conjunto está encendido y se desactive cuando se apaga.

Para esto se usa una salida de potencia de la SAI que abre/cierra un relé, manejando la interrupción/continuidad.

La lógica del circuito arriba propuesto es:

- Contacto abierto: Parada del SAI.
- Contacto cerrado: Puesta en marcha del SAI.

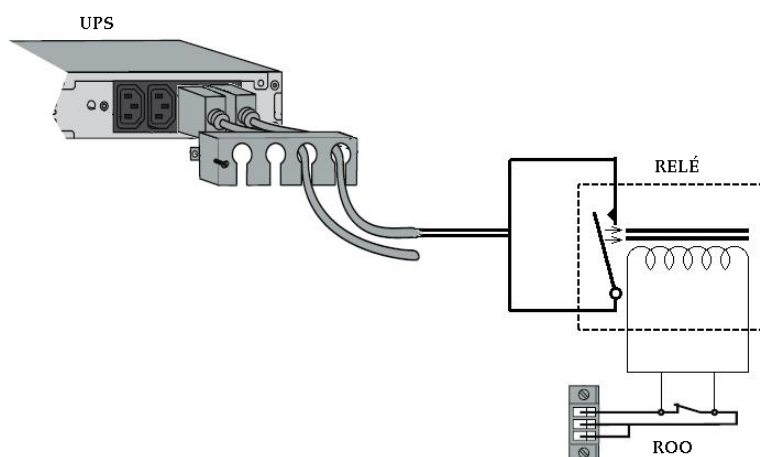


Figura 47 Lógica de marcha/parada de SAI's auxiliares.

1. De la SAI se obtiene la salida de potencia que activará el relé.
2. El relé abrirá/cerrará cuando este inactiva/activa la SAI.
3. Cuando abre, las SAI's conectadas se apagan, y cuando cierra, las SAI's conectadas se encienden.

#### 5.4.3.2. Servidor



Este será el equipo encargado de la gestión de todo el sistema de apagado por debajo de la SAI. Es considerado nodo central en la “red” de apagado. Es el equipo directamente conectado a la SAI y es el encargado de ordenar al resto de equipos su apagado, a través de comandos RSH enviados por red.

Para el correcto funcionamiento debe estar conectado mediante conexión USB punto-a-punto al puerto de comunicaciones USB de la SAI. Para esto es necesario, dependiendo del caso:

- Cable USB
- Hub de conexión de varias entradas USB (dependiendo del uso del puerto USB de cada servidor) .
- Dependiendo de la distancia a la que estén servidor y SAI, podría ser necesario intercalar un hub USB autoalimentado. Hay que tener en cuenta, por lo tanto, que la distancia máxima de un cable USB es aproximadamente 4 m.

En el servidor será donde esté instalado el software NMP, provisto por MGE, para el control de las SAI de esta marca. Dependiendo de la marca de la SAI quizá sea necesario modificar la configuración o estudiar algún diseño alternativo, ya que este sistema está ideado con las posibilidades que ofrecen las SAI inteligentes de MGE, presentes en todos los bancos. Además, aunque no ha sido necesario probarlo, el software de MGE también está disponible para Linux, así que el servidor podría estar perfectamente montado con Linux.

También hay que tener instalado un cliente RSH. Esto, en principio, viene por defecto en cualquier equipo con Windows 2000 o Unix, por lo que no será necesario realizar ninguna operación adicional.

Este cliente RSH es necesario para ejecutar el *script* de apagado configurado en el servidor, y que se explicará con más detalle más adelante.

#### 5.4.3.3. Script

#### 5.4.3.4. Configuración

Una vez que esté completada la instalación de NMP se crearán los iconos de acceso, y ya se puede entrar a configurar la SAI con dos opciones posibles:

- <https://<@IP>/mergeSAI2>
- A través de las opciones avanzadas del propio software.

Lo que se ve al entrar en el servidor Web es la siguiente página de inicio. A partir de aquí se pueden configurar los diferentes parámetros, a través del menú mostrado a la izquierda.

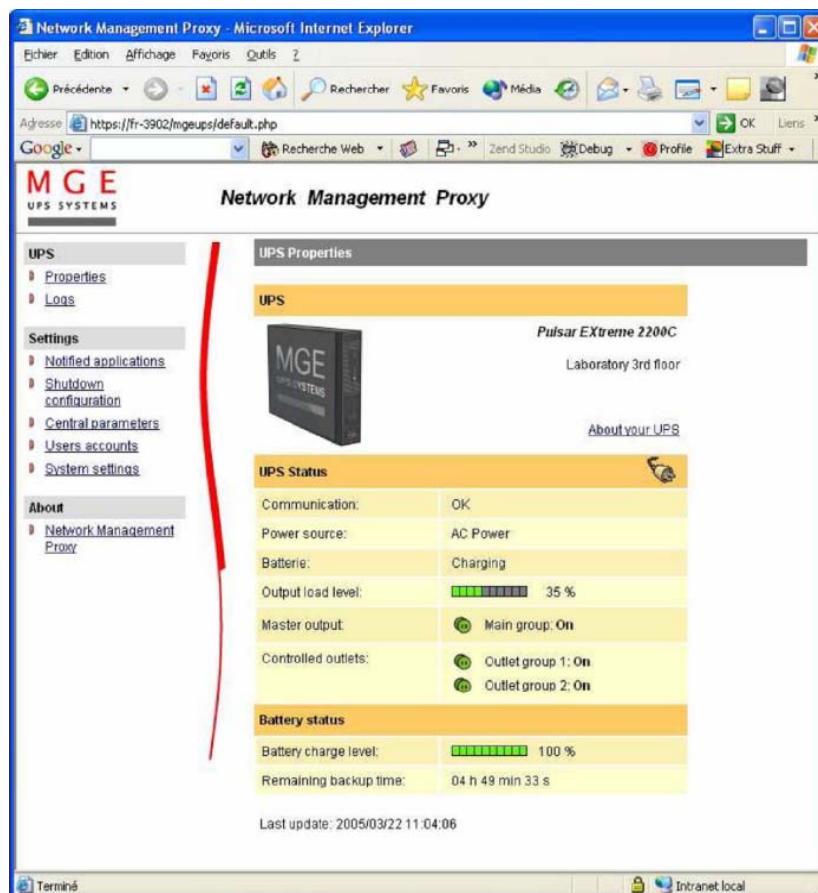


Figura 48 NMP Web Page

En la configuración de la SAI hay que modificar algunos parámetros relevantes.

a. Shutdown

Permite elegir el momento en que la SAI comenzará el proceso de *shutdown* después de haber perdido la alimentación. Se puede elegir que comience pasado un tiempo (*after*), que comience con un nivel de batería inferior a un valor (*if capacity under*) o cuando el tiempo restante sea menor que un valor (*if remaining time under*).

También hay que modificar la duración del apagado del *proxy*. Este será el tiempo de duración del apagado del servidor *proxy*. No es un tiempo fijo, ni definido, ni determinista, es un tiempo que se debe estimar. Hay que tener en cuenta, que la suma entre el tiempo de *shutdown* de la SAI, el tiempo de apagado del *proxy*, y el tiempo en que el *proxy* comenzará el apagado; no debe superar el tiempo de vida máximo a máxima carga de la SAI. Este tiempo se podrá encontrar en cada SAI concreta instalada en cada banco.

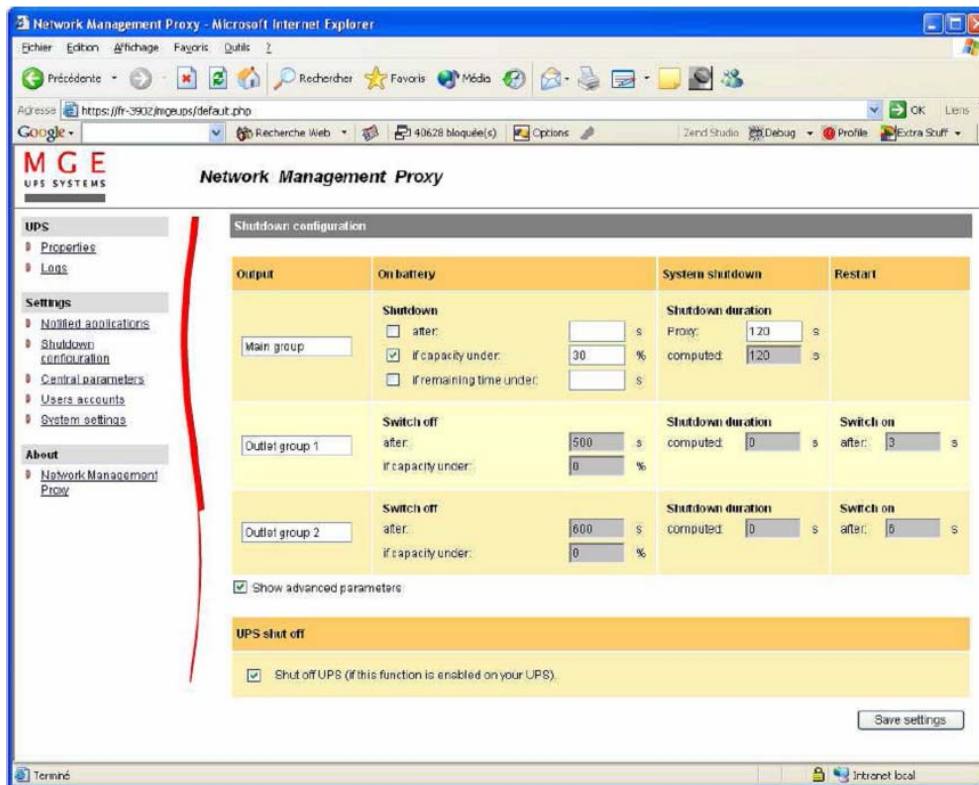


Figura 49 NMP Shutdown Configuration

En las opciones de *Central Parameters* se pueden modificar las opciones pertenecientes a los NSM. En este caso no está instalado este software instalado, por lo que no es necesario modificar las opciones.

Ahora hay que editar el *script* que se ejecutará cuando se inicie la rutina de apagado. En este *script* se deben añadir las directivas de apagado dependiendo del tipo de equipo.

```
rsh IP -l root "/opt/sudo/bin/sudo /sbin/reboot -h -m powershutdown"
```

```
shutdown \hostname\IP /C
```

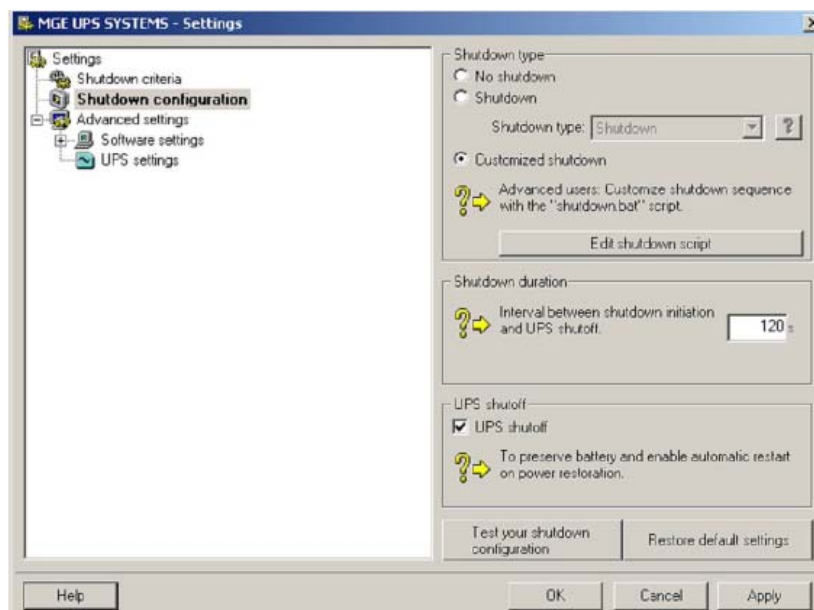


Figura 50 MGE SAI SYSTEMS - Settings

Se elige la configuración “*Customized shutdown*” y edita el *script*. Se añaden las directivas de apagado antes de la directiva ya escrita:

*Shutdown.exe –shutdown*

Se añaden las instrucciones antes, ya que en caso contrario no dará tiempo a ejecutarse. Generalmente es conveniente escribir las directivas poniendo primero *start cmd /c* para que todas las directivas se ejecuten en paralelo sin tener que esperar a que cada una acabe por su parte.

#### 5.4.3.5. Ejemplo de Shutdown

1	En el momento en que se detecta la pérdida de potencia, la SAI se lo indica al servidor, Management Proxy.
2	<p>El Management Proxy monitoriza los criterios de shutdown. Se lanza el shutdown cuando:</p> <ul style="list-style-type: none"> <li>• El tiempo de retardo de apagado se acaba</li> <li>• La batería baja por debajo de un nivel estipulado</li> <li>• El tiempo restante de la batería de la SAI baja por debajo de</li> </ul>

	un nivel <ul style="list-style-type: none"> <li>El tiempo restante de la batería de la SAI baja por debajo del tiempo estimado de apagado de los equipos.</li> </ul>
3	Cuando el servidor comienza el apagado ejecuta un script de apagado del resto de equipos.
4	En cuanto el tiempo estimado de apagado del proxy se consume, se apaga la SAI.
5	En el momento que se reponga la corriente, se volverán a encender todos los equipos dependientes de la SAI.

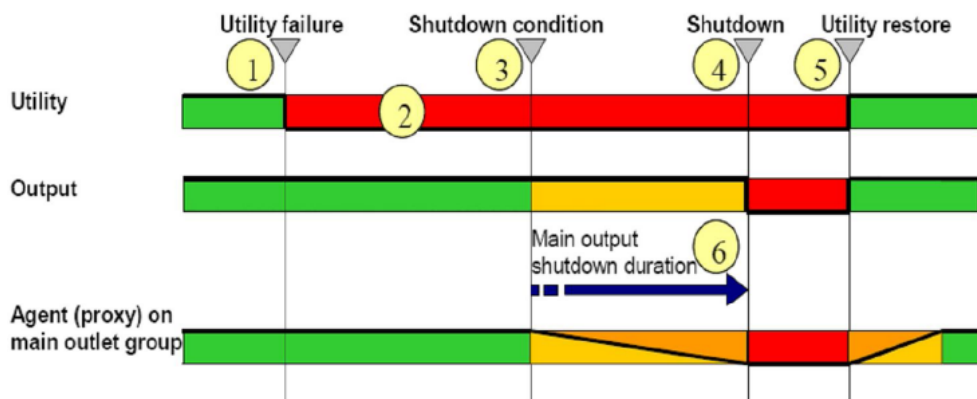


Figura 51 Ejemplo de shutdown [51]

#### 5.4.3.6. Corte de alimentación

Una ventaja que presentará la configuración que incluye SAI será la de evitar el apagado repentino de los equipos críticos de un banco. Al realizar un sistema de apagado ordenado partiendo de la SAI, ésta realiza esta tarea independientemente del motivo que la origine. Si el motivo es apagado remoto mediante el RPC, apagado presencial mediante el botón de apagado o corte repentino de suministro eléctrico, el sistema actuará igual, apagando ordenadamente todos los equipos.

Por lo tanto no sólo se consigue el objetivo inicial, si no que además se facilita la protección del equipo, ahorrando recursos económicos y de trabajo

significativos, ya que no será necesario realizar reparaciones de equipos, ni repetición de trabajo.

#### **5.4.4. Resumen de mejoras**

- I. **Sencillez en el apagado.** Con la nueva utilidad se facilita la tarea de apagado a pulsar un botón, ya sea mediante red o físicamente en el banco. Ahora se tiene la seguridad de que el banco quedará correcta y completamente apagado sin necesidad de realizar el apagado manual de toda la estructura de banco.
- II. **Seguridad de apagado de equipos sensibles.** Los equipos sensibles al apagado (basados en UNIX o Windows 2000) pueden producir errores tanto en los discos duros como en el SW del SO. Realizando un apagado ordenado y correctamente estructurado se tiene la seguridad de que todo seguirá funcionando bien. La cantidad de recursos, tanto económicos como temporales, perdidos debido a la necesidad de sustitución de HW o SW por estos errores se ahorra con la nueva funcionalidad.
- III. **Ahorro económico.** El gasto producido tanto por las pérdidas anteriores, como por el mantenimiento del banco permanentemente encendido se calculó previamente. Con la mejora propuesta se consigue ahorrar gran cantidad de potencia consumida por la instalación.

#### **5.5. Funcionalidad conjunta de las mejoras propuestas**

Las mejoras propuestas tanto para el subsistema de ensayo como para el módulo de comunicaciones, son mejoras relacionadas. Sin el nuevo sistema de gestión de versiones sería imposible manejar la flexibilidad de configuraciones que se facilita con la inclusión de las matrices de conmutación. De hecho la necesidad de incluir las matrices es la que induce la necesidad de obtener un sistema más robusto de manejo de ficheros de configuración.

Con las matrices de conmutación, se puede emplear un banco para varios proyectos con tan sólo modificar la estructura SW. Actualmente, un banco puede disponer de varias configuraciones, pero generalmente suele ser un grupo reducido de personas el que lo modifica. Sin embargo, incluyendo varios proyectos en un solo banco, son varias personas, en varios grupos, trabajando en paralelo en la configuración. Sin disponer de un sistema como el propuesto, sería inmanejable la compresión de las configuraciones disponibles en los múltiples ficheros del banco.

La mejora propuesta para el módulo de alimentación no está directamente ligada a las otras mejoras, pero ayudará a mantener la seguridad ante fallos de alimentación en los nuevos diseños de banco. Si alguno de los equipos introducidos (matrices u otros) son sensibles al apagado brusco, se pueden incluir en la lista de equipos para apagar de manera segura, facilitando el trabajo de todo el banco.



## **Capítulo 6. Conclusiones**

Para una correcta realización del proceso de optimización del banco de ensayo, es necesaria una primera fase de entendimiento de la estructura y herramientas empleadas en el banco de ensayo. Este proceso previo permite una completa comprensión de todo el entorno de pruebas diseñado para la realización de test.

Se ha conseguido la comprensión del concepto de banco de ensayo, así como su uso y propiedades más importantes. Aunque no representaba el objetivo directo de éste proyecto, es una parte importante, ya que para comprender el objetivo principal, es necesario conocer el entorno completo.

La línea de trabajo seguida ha garantizado un correcto desarrollo de los sistemas necesarios para conseguir la optimización de cada una de las partes del banco de ensayo. Se ha conseguido la optimización de los aspectos más importantes, consiguiendo mayor modularidad y un sistema más abierto y completo que lo encontrado al inicio, llegando a comprender cuál es la línea por la que actualmente están evolucionando los bancos de ensayo.

Esta línea de progresión pretende conseguir crear una nueva generación de bancos de ensayo basados en mejorar los conceptos previos y añadir nuevos, y este trabajo ayuda a esta tarea futura de mejora mediante el aprendizaje e implementación de soluciones factibles y completas.

La tarea de optimización y mejora de todos los aspectos relevantes de un banco de ensayo es una tarea larga y complicada. Sin embargo con este proyecto se ha contribuido, en una parte importante, a mejorar aspectos fundamentales en cada uno de los módulos del banco.

Aunque es un buen punto de comienzo, el proceso continuará en los próximos años, ayudando a mejorar tanto local como internacionalmente las instalaciones y repercusión de los bancos de ensayo en el entorno de las pruebas en tierra de aviónica.

Por lo que el objetivo global que se persigue y que con este proyecto se ha contribuido, es ser una referencia mundial en la creación y mantenimiento de una nueva generación bancos de ensayo orientados a las pruebas en tierra sobre sistemas de aviónica.

Se puede concluir, por lo tanto, que se ha conseguido el objetivo inicial del proyecto.

### **6.1. Trabajos Futuros**

Los trabajos futuros en el campo de los bancos de ensayo se están orientando hacia sistemas multiprograma integrados en sistemas flexibles capaces de soportar las diversas estructuras existentes.

Tanto el subsistema de ensayo como el módulo de comunicaciones deben rediseñarse para garantizar el soporte de bancos virtuales contruidos a partir de diversas partes instaladas de manera genérica. Por lo tanto también es necesaria la optimización del modulo de alimentación para que garantice la capacidad de ampliación y modificación. Estos bancos virtuales permitirán la utilización de los bancos en múltiples proyectos con una buena gestión de la configuración y obtención de datos. Por lo tanto, no será necesaria la construcción de un banco para cada proyecto, si no que se podrá emplear un único subsistema de ensayo en la implementación de múltiples programas.

La instrumentación de adquisición de datos, como se estudió en este proyecto, se está orientando hacia sistemas basados en red, sin una necesidad estricta de mantenerse cerca de los sistemas analizados. En este punto los sistemas LXI y el desarrollo de buenos sistemas de minería de datos se hace fundamental para garantizar tanto la integridad como el análisis.

Los objetivos que se pretenden conseguir en los próximos años se pueden entender mejor analizando los objetivos en el banco por partes:

## **1. Subsistema de ensayo**

- a. Adaptar el sistema a los nuevos procesadores *multicore*.
- b. Completa flexibilidad de configuración.
- c. Facilitar la interacción con el usuario.
- d. Integración con nuevas tecnologías de instrumentación (LXI...)
- e. Seguridad e integridad de los datos.

## **2. Módulo de comunicaciones**

- a. Flexibilidad de instalación.
- b. Facilidad en la construcción.
- c. Completa modularidad del sistema.
- d. Apertura del sistema a las nuevas tecnologías.
- e. Mejora en la detección de errores.

## **3. Módulo de alimentación**

- a. Adaptación del consumo a las exigencias actuales.
- b. Mejora en la seguridad personal del banco.
- c. Mejora en la seguridad sobre los equipos del banco.
- d. Flexibilidad de inclusión de nuevos equipos.

## **Anexo I.     Descripción de las herramientas del subsistema de ensayo**

### **AI.i.       Introducción**

Se va a proceder a realizar una descripción más completa de las herramientas desarrolladas para el subsistema de ensayo SEAS.

Se van a definir las siguientes herramientas:

- Graphical Data Visualization System.
- Recording.
- Análisis.

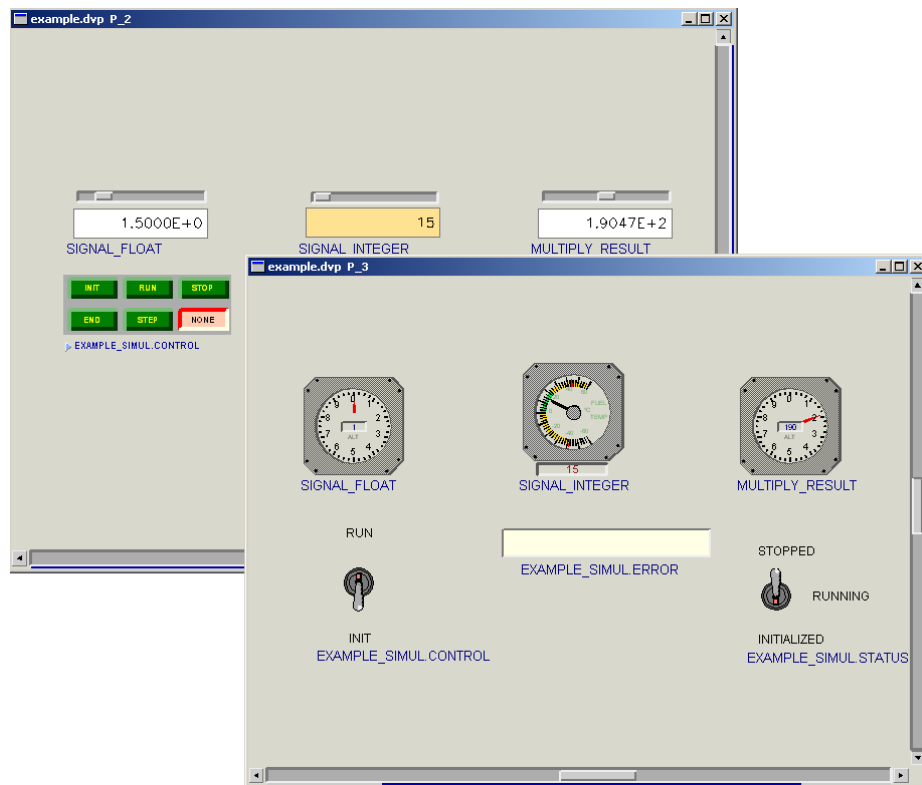
Todas ellas están disponibles actualmente en el subsistema de ensayo y siguen en continua mejora, por lo que para conocer el estado actual será necesario referirse a los desarrolladores de la empresa.

### **AI.ii.     Graphical Data Visualization System**

#### **I.       Visualización**

La visualización [20] de las señales será muy flexible para que cada usuario tenga la capacidad de adecuarlo a sus necesidades. Se dispone de gran cantidad de *widgets*, permitiendo simular si fuera necesario una cabina de avión.

El sistema de pruebas no se empleará sólo en pruebas de equipos si no que se puede necesitar simular un sistema, para lo que se necesitan miles de señales. Representar todas estas señales en un panel plano con los valores puros de las señales sería imposible, por lo que disponer de *widgets* usuales de aviación es realmente útil.



**Figura 52 Composición de Paneles GVDS**

GDVS facilita representaciones por defecto de cada señal dependiendo de su tipo, pero en general, salvo para casos muy sencillos, se quiere adecuar la representación. Además existe la opción de crear nuestro propio *widget*, dando una capacidad indefinida de representación de las señales de una simulación.

## II. Estimulación

GDVS no sólo aporta la capacidad de simulación y visualización de señales, si no que es capaz de estimular señales.

Tiene la capacidad de definir señales que se estimularán mediante algunos tipos predefinidos:

- Cuadrada: Será una señal con 2 valores definidos, un valor alto Y1 que estará activo un tiempo T1, y un valor Y2, que estará activo un

tiempo T2. En general serán valores distintos, y cuando finaliza el periodo T2, comienza de nuevo con Y1.

- Senoidal: Se estimulará una señal senoidal con período T, valor medio Y1 y amplitud.
- Triangular: Se define una señal con un valor Y1 inicial y un valor Y2 final, y los períodos de duración de cada tránsito. El tránsito de Y1 a Y2 y viceversa no tienen porque tener el mismo valor. Si se define alguno de ellos a 0, se creará un diente de sierra.
- Rampa: Debemos definir un periodo para la señal, que realizará la variación entre un valor Y1 y un valor Y2.
- Aleatoria: Definimos un valor MAX y un valor MIN, y se generarán números aleatorios entre estos 2 valores.

### III. Representación

La representación de las señales no siempre se quiere que sea la misma. Dependiendo del tipo de señal y de la utilidad de esta, habrá que representarlo en el formato adecuado.

Lo normal será que el valor esté escalado, puesto que, en general, es necesaria la representación de muchas señales, y lo importante es que la información quede bien representada en un golpe de vista.

El valor escalado de la señal se representará como:  $Y=f(x)$ , siendo Y el valor final, y x el valor plano adquirido.

Los tipos de valores escalados predefinidos pueden ser:

- Hexadecimal
- Binario
- Octal
- Decimal

Además de haber algunos valores de escalado predefinidos, también tiene la posibilidad de aplicar fórmulas de ingeniería que expliquen mejor el concepto de la señal.

Los formatos y representaciones anteriores son aptos para tipos numéricos de señales, pero también está definido el tipo ENUM. Este tipo tiene definidos un conjunto de valores que puede tomar la señal, por lo que será necesaria la representación y estimulación adecuada de éstos.

#### IV. Simulaciones Gráficas

En muchos casos, las simulaciones de escenarios de pruebas deberán incluir una representación gráfica que debe estar conectada a las señales de la simulación. Este sería el caso por ejemplo de una simulación Radar.

#### AI.iii. Recording

##### I. Añadir señales

Cada *recording definition* [21] está asociado a un único test y a una base de datos de señal, así que las señales añadidas al esquema sólo pertenecerán a este test.

Al principio no habrá ninguna señal disponible, así que hay que buscarlas en la base de datos. Ya se conoce la magnitud de la base de señales, por lo que será útil buscarla por nombre, equipo al que pertenece I/O, etc.

Algunas de las señales que en general siempre estarán incluidas, son las señales de la propia aplicación de *recording*. Estas señales están disponibles para el manejo automático de *Start/Stop* e inicialización.

##### II. Editar señales

Una vez que se han elegido todas las señales necesarias, se pasa a editar la frecuencia de grabación de las señales. Por defecto estará establecida en 125 Hz, pero se puede ajustar a lo que se desee.

En la edición de las señales se incluye la posibilidad de eliminar señales de la definición. La definición del *recording* se puede grabar, por lo que ante una prueba posterior, quizá haya señales no necesarias, o quizá sea necesario añadir algunas más.

### III. Ejecutar recording

Para cada *recording definicion*, el sistema crea 3 señales de control:

- RecordingDefinition.Control: Señal usada por el usuario para controlar la simulación, INIT, EXEC, STOP y END.
- RecordingDefinition.Status: Señal que indica al usuario sobre el estatus de la grabación, INITIALIZED, URNG, STOPPED, ENDED y DEBUT.
- RecordingDefinition.Error: Empleada para indicar al usuario cualquier error sucedido.

Hay dos posibilidades de inicio del *recording*. Ambas comienzan de la misma forma, pero están basadas en distintas utilidades:

- A. Iniciar el cliente SEAS.
- B. Login en sistema.
- C. Abrir el correspondiente nodo.
- D. Elegir el test para ejecutar, o crear uno nuevo.
- E. Comenzar el test.

Dos posibilidades para ejecutar el test:



1. GDVT. A través de la utilidad de visualización y estimulación, permite crear un panel con las señales internas del *recording*, indicando manualmente el inicio de éste.
2. A través de otra utilidad, *simulation control*, que ya tiene definidas estas posibilidades. Se mostrarán las simulaciones que pueden ser gestionadas, entre las que se encontrará la de *recording*.

#### Al.iv. Análisis

El análisis de señales [23] puede tener diferentes apartados o necesidades. Habrá, en general, 3 análisis fundamentales:

- **Comprobación de entradas y salidas.** En caso de tener unas especificaciones concretas para las salidas del subsistema se puede realizar el análisis directo, chequeando que para un conjunto de entradas concretas, se obtengan unas salidas concretas. Este análisis es correcto en caso de sistemas modelados “deterministas”. Un modelo correctamente definido se podrá representar con números fijos, sin embargo un sistema real podrá representarse dentro de unos márgenes de valores, por lo que habría que decidir una tolerancia aceptable.
- **Comprobación de leyes físicas.** En los modelos realizados o en bancos se obtendrán un conjunto de salidas. Estas salidas pueden estar relacionadas mediante leyes físicas que dirigen el modelo, por lo que habrá que realizar un análisis matemático de los datos a través de un SW adecuado (Matlab...) y verificar que cumple las leyes teóricas.
- **Filtrado y comprobación de datos.** En caso de tener únicamente datos analógicos que a primera vista no representen lo que se busca, será necesario filtrar y buscar los datos necesarios. Si se busca una frecuencia de resonancia de algún elemento de avión, será necesario el muestreo de datos y posterior análisis en frecuencia.

El análisis de datos es un tema extremadamente amplio, que no es el objetivo de este apartado. El análisis final se realizará con las herramientas propias del subsistema de ensayo y otros programas de tratamiento de datos como Matlab.

## **Anexo II. Descripción de las herramientas de simulación SEAS**

### **All.i. Introducción**

Las simulaciones desarrolladas para el banco, deben tener unas características concretas que permitan la sencilla integración con el subsistema de ensayo del banco de integración. Estas características se presentan a continuación para los tipos de simulaciones permitidos.

- Simulaciones escritas en lenguaje ADA
- Simulaciones no escritas en ADA

### **All.ii. Simulaciones ADA**

A la hora de preparar una simulación [24], es necesario configurar diferentes partes del sistema:

#### **All.ii.a. Configuración**

##### **I. Definición de las señales de la simulación.**

La simulación necesitará manejar un conjunto de señales de entrada, salida o de entrada y salida para enviar o adquirir y manejar los datos que necesita. Pero también existirán muchas señales que emplearán otras simulaciones, o que tan sólo serán señales de control para el usuario. Todas estas señales es necesario que estén correctamente configuradas.

En esta definición de señales hay que describir las señales que necesarias, incluyendo una definición de tipos, equipos a los que pertenecen, rango y velocidad de muestreo.

Una vez que están definidas, el manejo será mucho más sencillo, puesto que en la simulación, con tan sólo referirse a un nombre, se hace referencia,

por ejemplo, a un rango de bits, dentro de un campo de una parte de mensaje Ethernet.

En teoría no hay límite para la cantidad de señales posibles en un nodo, aunque las limitaciones estarán en las velocidades y capacidades de los procesadores. Actualmente hay configuraciones con hasta 500.000 señales, e incluso se llega a una velocidad de 1 KHz en muestreo de las señales.

## II. Definición de parámetros.

Ahora hay que definir los parámetros exclusivos de la simulación.

Se debe definir el procesador en el que se ejecutará, en caso de que sea necesario, la tasa de repetición de la simulación, y sobretodo añadir ahora a la simulación, las señales que usaremos en ella.

Se pueden añadir todas las señales que se necesiten, independientemente al equipo que pertenezcan, puesto que posteriormente se realizará un mapeo de estas señales, definiendo su ubicación y su manejo.

Una vez que están añadidas las señales, se debe configurar o modificar ciertos aspectos de ellas.

- Cambiar el nombre en la simulación. Existe la posibilidad de tener un nombre de señal dentro del sistema, y tener otro nombre de señal en la simulación. Esto facilita el enlace entre la simulación y el sistema, ya que se podrá cambiar de señal sin cambiar la simulación.
- Modificar el modo Plano/Escalado. Se tiene la opción de introducir o adquirir el valor de la señal con un valor escalado o directamente plano.
- Modificar el tipo de la señal. Cuando se realiza la interfaz de definición de las señales en código, hay que definir el tipo de dato. Una cosa es el tipo o rango de la señal en el sistema, y otro es el

tipo que se tiene en la simulación. Aunque obviamente está relacionado se usará el tipo que se necesite.

- Modo entrada-salida. Se puede modificar el tipo de señal que usada; entrada, salida o entrada y salida, o incluso ninguna de ellas. En este último caso la simulación ni escribe ni lee estas señales, pero quizá necesita algunos parámetros de la señal.

### III. Generación de interfaz de simulación.

Una vez que todos los detalles del proceso están definidos y ajustados, se genera la interfaz de código de la simulación. Este será el código de definición de propiedades de las señales del sistema dentro de la simulación.

Se generará automáticamente en el directorio de la simulación, por lo que a la hora de compilar no es necesario realizar más modificaciones.

La interfaz garantiza una forma sencilla de comunicación entre la simulación y el sistema, a través de la definición de las señales de simulación. Sin este código no sería posible enlazar la simulación con las señales de ésta.

#### **All.ii.b. Compilación**

Antes de la primera compilación de la simulación, incluyendo código generado por el usuario e interfaz de señales, es necesario crear varios paquetes de ficheros.

#### I. Ficheros de proceso

Se crean dos ficheros, preferentemente en ADA:

- SimName\_process.ads
- SimName\_process.adb

Estos paquetes reúnen algunas acciones de control necesarias para las simulaciones, aunque el grueso del paquete deberá ser modificado por el usuario para integrar lo que necesite. En estos paquetes se implementan las funciones INIT, EXEC y STOP.

- **INIT:** Función de inicialización empleada cuando se quiere iniciar la simulación. Hay que llamar al proceso INIT\_Inter para inicializar todos los objetos asociados a la simulación. Además de llamar otras subrutinas necesarias.

```
procedure INIT (RATE : in NATURAL;
               Simul_Status : out Hstat.Status_Type ) is
begin
    simul_v31_Interf.Init_Interf;
    Simul_Status := Hstat.Hsimul.Ok;

    -- TO COMPLETE (IF NEEDED)-----
    -----
exception
    when others =>
        Simul_Status := Hstat.Set_Status(Hstat.Fatal,
                                         "HSIMUL",
                                         "Excepcion en INIT simul_v31");
end INIT;
```

Figura 53 Proceso INIT de Simulación

- **EXEC:** Es la rutina principal. Se llama cada vez que el *scheduler* despierta al proceso de acuerdo con su tasa de ejecución. En general este proceso llamará a una función Get\_Values, procesará los datos y llamará a Put\_Values, para escribir lo que necesite.

```
procedure EXEC (Simul_Status : out Hstat.Status_Type ) is
begin
    -- TO DO (UNCOMMENT) -----
    --
    --simul_v31_Interf.get_values;
    --simul_v31_Interf.put_values;
    --
    -----

    Simul_Status := Hstat.Hsimul.Ok;

exception
    when others =>
        Simul_Status := Hstat.Set_Status(Hstat.Fatal,
                                         "HSIMUL",
                                         "Excepcion en EXEC simul_v31");
end EXEC;
```

Figura 54 Proceso EXEC de Simulación

- STOP: Es la función de finalización, que se llama cuando el usuario quiere finalizar la ejecución de la simulación. En este punto es necesario llamar a los procedimientos de parada de la simulación y liberar memoria.

```

procedure Stop (Simul_Status : out Hstat.Status_Type ) is
begin
  -- TO COMPLETE (IF NEEDED)-----
  -----
  Simul_Status := Hstat.Hsimul.Ok;

  exception
    when others =>
      Simul_Status := Hstat.Set_Status(Hstat.Fatal,
                                      "HSIMUL",
                                      "Excepcion en STOP simul_v31");
end STOP;

```

**Figura 55 Proceso STOP de Simulación**

## II. Ficheros de sistema

Los siguientes ficheros se crean siempre que se compile la simulación. No se deben hacer modificaciones, ya que se crearán siempre que se compile.

- SimulName\_Main.adb. Es la rutina principal de la simulación.
- Signal\_process\_Cimulation.adb. Se encarga de la programación de tareas del proceso, leer ficheros...
- Link\_Dll.ads. Es un paquete específico que importa las librerías necesarias para el sistema dependiendo del tipo de nodo (VME, PXI o ambas).

### All.ii.c. Ejecución de la simulación

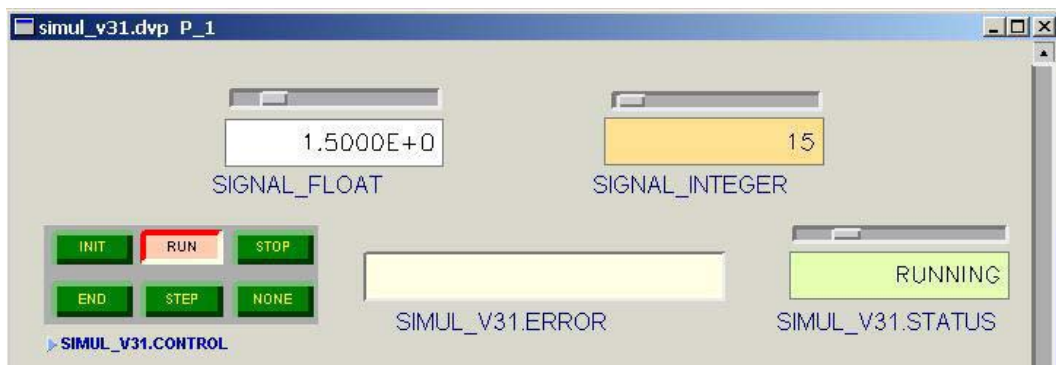
Para ejecutar cualquier simulación, lo primer oque se necesita es tener privilegios de ejecución de test en SEAS. Existen 3 señales de control creadas para cada simulación definida.

- SimulationName.Control: Empleada para indicar a la simulación la función a ejecutar, INIT, EXEC, STOP, END o STEP.

- `SimulationName.Status`: Indica el estado en que se encuentra la simulación, INITIALIZED, URNG, STOPPED, ENDED y DEBUG
- `SimulationName.Error`: Esta es la señal que se utiliza para notificar cualquier mensaje de error.

Se dispone de dos formas de ejecutar la simulación.

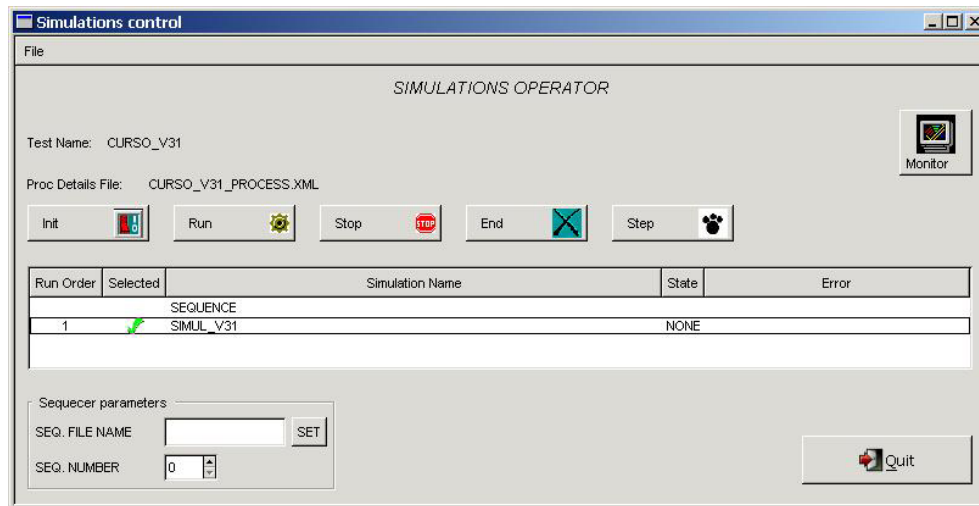
- GVDT. A través del entorno gráfico de visualización y estimulación, se puede indicar con la señal `SimulationName.Control` la inicialización, ejecución, parada... de la simulación.



**Figura 56 Pantalla de control de simulación por GVDT**

- Simulation Control. En la aplicación *simulation control*, se dispone de las mismas posibilidades que antes. Es una aplicación gráfica definida para la realización exclusiva de operaciones sobre las simulaciones.





**Figura 57 Pantalla de control de simulación por Aplicación**

#### **All.ii.d. Debugging**

Para la depuración de una simulación se usará, generalmente, el entorno GPS.

Durante la compilación, se crea una conexión virtual con una carpeta compartida. La sesión de depuración se debe ejecutar en el nodo servidor, donde la simulación corre. Además, se necesita un cliente para poder iniciar, ejecutar y parar la simulación.

#### **All.iii. Simulaciones Non Ada**

Como se indicó para las simulaciones en ADA, los puntos más importantes que hay que tener siempre, son las funciones de inicialización, ejecución y parada de la simulación.

Para integrar la simulación, hay que crear un binding package donde se declaran las funciones y variables exportadas por el ".dll".

Un código de ejemplo de importación de funciones en C++ hacia ADA:

#### **I. C Header File**

```

#ifdef __cplusplus
extern "C" {
#endif

#ifdef DIVIDE_EXPORTS
#define DIVIDE_API __declspec(dllexport)
#else
#define DIVIDE_API __declspec(dllimport)
#endif

extern DIVIDE_API int nDivide;
DIVIDE_API float start ();
DIVIDE_API void execute (int *, float *);
DIVIDE_API float finish (int);

#ifdef __cplusplus
}
#endif

```

## II. C Source File

```

#include "stdafx.h"
#include "divide.h"
//Defines the entry point for the DLL application.
BOOL APIENTRY DllMain( HANDLE hModule,
                      DWORD ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    switch (ul_reason_for_call)
    {
        case DLL_PROCESS_ATTACH:
        case DLL_THREAD_ATTACH:

```

```

        case DLL_THREAD_DETACH:
        case DLL_PROCESS_DETACH:
            break;
    }
    return TRUE;
}

// This is an example of an exported variable
DIVIDE_API int nDivide=0;
// Exported functions with no parameters.
DIVIDE_API float start ()
{
    nDivide = 145;
    return float(8.12);
}

// Exported function with input/output parameters (passed as
//reference).
DIVIDE_API void execute (int *v_integer, float *v_float)
{
    v_integer[0] = (v_integer[0]++) % 1000;
    v_float[0] = float(v_integer[0]) / 100;
}

// Exported function with an input parameter and returns a value
//(passed as value).
DIVIDE_API float finish (int v_integer)
{
    return float(v_integer) / 300;
}

```

### III. ADA Binding Source File

```

package DIVIDE_C is
    -- Variable imported from Dll (nDivide)
    C_Integer : Integer;
    -- Functions imported from Dll (nDivide)

```

```

function START return float;
procedure EXECUTE (Sig_Integer : access integer;
Sig_Float : access float);
function FINISH (Sig_Integer : integer) return float;
private
    -- Association between Ada functions and DLL exported ----
    -- Functions
    pragma Import ( StdCall, C_Integer, External_name =>
    "nDivide");
    pragma Import ( C, START, External_name => "start");
    pragma Import ( C, EXECUTE, External_name =>
    "execute");
    pragma Import ( C, FINISH, External_name => "finish");

end DIVIDE_C;

```

#### IV. ADA Simul Process Source File

```

with Hstat.Hsimul;
with simul_v31_Interf;
with DIVIDE_C;

package body simul_v31_PROCESS is
    -----
    -- Procedure_Name : INIT --
    -- ----- --
    -- --
    -- It MUST call procedure Init_Interf to initialize --
    -- all signal Objects associated to the simulation. --
    -- --
    -- It should call all other initialization routines --
    -- of the simulation. --
    -----

    procedure INIT (RATE : in NATURAL;
    Simul_Status : out HStat.Status_Type ) is

```

```

begin
    simul_v31_Interf.Init_Interf;
    -- Call To START Function.
    Simul_V31_Interf.Signal_Float := DIVIDE_C.START;
    Simul_V31_Interf.Signal_Integer :=
    Divide_C.C_Integer;
    -- Put values of the signals modified in START
    Function.
    simul_v31_Interf.Put_values;
    Simul_Status := Hstat.Hsimul.Ok;
exception
    when others =>
        Simul_Status := Hstat.Set_Status(Hstat.Fatal,
        "HSIMUL", "Excepcion en INIT simul_v31");
end INIT;

-----

-- Procedure_Name : EXEC --
-- -----
-- --
-- It should call Get_values, process the values and --
-- write new values if needed --
-----

procedure EXEC (Simul_Status : out Hstat.Status_Type ) is
    V_Int : aliased Integer;
    V_Float : aliased Float;
begin
    -- Get signals values
    simul_v31_Interf.get_values;
    V_Int := Simul_V31_Interf.Signal_Integer;
    V_Float := Simul_V31_Interf.Signal_float;
    -- Call To EXECUTE Function.
    DIVIDE_C.EXECUTE (V_Int'Access,
    V_Float'Access);

```

```

        Simul_V31_Interf.Signal_Integer := V_Int;
        Simul_V31_Interf.Signal_float := V_Float;
        -- Put signals values.
        simul_v31_Interf.put_values;
        Simul_Status := Hstat.Hsimul.Ok;
exception
    when others =>
        Simul_Status := Hstat.Set_Status(Hstat.Fatal,
        "HSIMUL",
        "Excepcion en EXEC simul_v31");
end EXEC;

-----

-- Procedure_Name : STOP --
-- -----
-- --
-- It should call Stop procedures of the simulation in --
-- order to free memory allocated during the execution. --
-----

procedure Stop (Simul_Status : out Hstat.Status_Type ) is
begin
    -- Call To FINISH Function.
    Simul_V31_Interf.Signal_Float := DIVIDE_C.FINISH
    (simul_v31_Interf.Signal_Integer);
    -- Put signals values.
    simul_v31_Interf.put_values;
    Simul_Status := Hstat.Hsimul.Ok;
exception
    when others =>
        Simul_Status := Hstat.Set_Status(Hstat.Fatal,
        "HSIMUL",
        "Excepcion en STOP simul_v31");
end STOP;
end simul_v31_PROCESS;

```

#### **All.iii.a. Compilación**

Aunque ahora hay algunas cosas añadidas sobre esta simulación, la compilación se lleva a cabo de la misma forma que antes. La única diferencia es que ahora se debe añadir la librería “.dll” como recurso de compilación

#### **All.iii.b. Ejecución de la simulación**

Igual que para la compilación, ahora no existe diferencia sobre la ejecución.

En este caso es más fácil de entender, ya que la ejecución se realiza sobre objetos de ejecución, no sobre código, por lo que si existen las funciones necesarias INIT, EXEC y STOP, independientemente del lenguaje o del método de compilación, se podrá ejecutar del mismo modo.

### **Anexo III. Descripción de buses de aviónica**

#### **AIII.i. Introducción**

Es necesario especificar las características más importantes de los diferentes buses de comunicaciones empleados en aviónica. Las características técnicas de cada bus se presentan a continuación de una manera teórica y más completa.

- ARINC429.
- AFDX.
- MIL-STD-1553
- STANAG3910
- CAN

#### **AIII.ii. Arinc 429**

### **III. Capa física**

ARINC429 [28], [29] usa par trenzado apantallado para la transmisión de las palabras de datos (32 bit). Entre cada transmisión se establece un tiempo de guarda de 4 bit, evitando tener que transmitir una señal de sincronismo en cable o en trama/supertrama.

La transmisión se realiza de forma diferencial entre cada cable del par trenzado. De los dos cables, uno se especifica como HI y otro como LO. Se emplea una modulación bipolar de retorno a cero (BPRZ), por lo que cada cable tendrá un rango de  $\pm 6V$ , obteniendo 3 rangos:

- Nivel Alto (HI): La diferencia de voltaje (A-B) estará entre 7.25V y 11V.
- Nivel Cero (NULL): La diferencia (A-B) debe estar entre 0.5V y - 0.5V.



- Nivel Bajo (LO): La diferencia será la opuesta de HI, -7.25V y -11V.

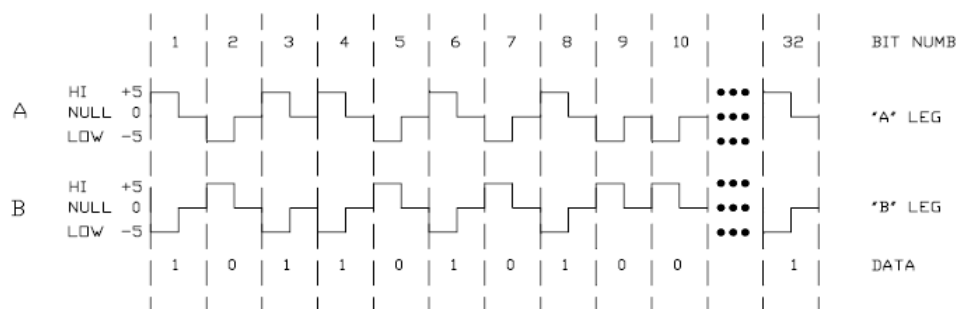
El “1” lógico se envía con un pulso HI seguido de un pulso NULL. Un “0” lógico se consigue enviando pulso LO seguido de pulso NULL. El período de cada nivel es siempre el mismo, igual que los tiempos de subida y bajada, dependiendo de la velocidad que se quiera conseguir. Se obtienen dos velocidades de transmisión, 100Kbps y 12.5Kbps.

La siguiente tabla resume las propiedades. Los tiempos de subida y bajada se establecen entre el 10% y 90% de valor final de la señal.

Parameter	High Speed	Low Speed
Bit Rate	100K bits/second	12.5K-14.5K bits/second
Time Y (one bit)	10 $\mu$ sec $\pm$ 2.5%	1/(bit rate) $\mu$ sec $\pm$ 2.5%
Time X	5 $\mu$ sec $\pm$ 5%	Y/2 $\mu$ sec $\pm$ 5%
Pulse Rise Time	1.5 $\pm$ 0.5 $\mu$ sec	10 $\pm$ 5 $\mu$ sec
Pulse Fall Time	1.5 $\pm$ 0.5 $\mu$ sec	10 $\pm$ 5 $\mu$ sec

**Figura 58 Características ARINC429**

Esta modulación es de menor potencia que una modulación de no retorno a cero, pero tiene el inconveniente de transferencia de energía cuando se envían cadenas largas con el mismo valor (HI o LO), debido a que se consigue un valor medio distinto de 0. Por lo tanto, los transmisores deben tener una impedancia de salida de  $75 \pm 5 \Omega$  y deben mantener un cortocircuito entre los cables y/o a tierra.



**Figura 59 Modulación ARINC429**

El voltaje recibido en cada par depende de la distancia a los receptores y del número de éstos, no más de 20 por par. Cada par es *simplex*, por lo que todo aquel que quiera transmitir y recibir debe disponer de dos pares. En cada par sólo existe un transmisor, por lo que al menos existirán tantos pares como transmisores haya en el sistema.

Es un protocolo de envío de mensajes únicos de 32 bit, por lo que puede transmitir una sola palabra. El equipo estará siempre transmitiendo un mensaje o transmitiendo NULL.

Electrical Characteristic	Value
Voltage Levels, each leg with respect to ground	+5V, 0V, -5V
Voltage Levels, Leg A with respect to Leg B	+10V, 0V, -10V
Bit Encoding	Bipolar Return to Zero
Word size	32 bits
Bit Rates	100K or 12.5K bits per second
High Speed Slew Rate	1.5 +/- 0.5 $\mu$ sec
Low Speed Slew Rate	10 +/- 5 $\mu$ sec

**Figura 60 Características eléctricas ARINC429**

#### IV. Formato de Mensaje

El mensaje está compuesto típicamente de 32 bits, y numerado de la forma 1..32. Existen cinco campos principales:

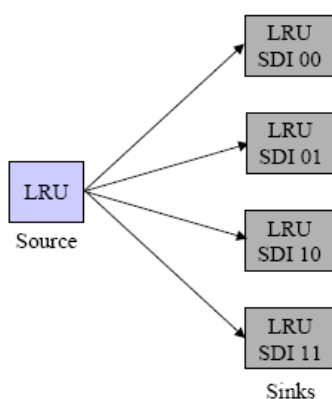
- a) Label: Indica el tipo de mensaje que se envía y los parámetros que se necesitan. Cada equipo tiene asignado un identificador según el tipo.

Cuando se realice el envío de bloques o transferencia de ficheros, este campo se reemplaza por **System Address Label (SAL)**.

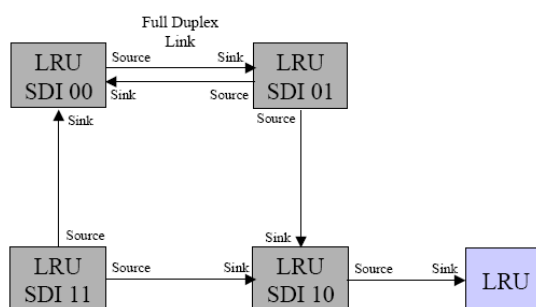
- b) Source/Destination Identifier: Se emplea para diferenciar el receptor de un mensaje, para distinguir el emisor del mensaje en caso de

sistemas multi transmisor y algunas veces como extensión de datos o de Label.

En las siguientes figuras se muestran las combinaciones en caso de usarse como identificador de receptor o identificador de transmisor.



**Figura 61 Configuración 1 ARINC429**



**Figura 62 Configuración 2 ARINC429**

- c) Data: Es el campo en el que se envía la información que se necesita. Hay diversos modos de envío de la información:
- BCD. Se usan 4 bits para la representación de cada cifra decimal. Al ser 19 bits de datos, uno de los caracteres tendrá 3 bits, por lo que, si el valor debe ser mayor de 7 se convertirá automáticamente en *padding* y el primer carácter pasará a ser el segundo.

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	1
P	SSM		CHAR 1			CHAR 2			CHAR 3			CHAR 4			CHAR 5			SDI			LABEL				

**Figura 63 Ejemplo trama BCD ARINC429**

- BNR. El número a transmitir se almacena en forma binaria del mismo modo que cualquier máquina común. El bit “29” se usa como bit de signo, por lo que se consigue un rango de  $\pm 262143$ .

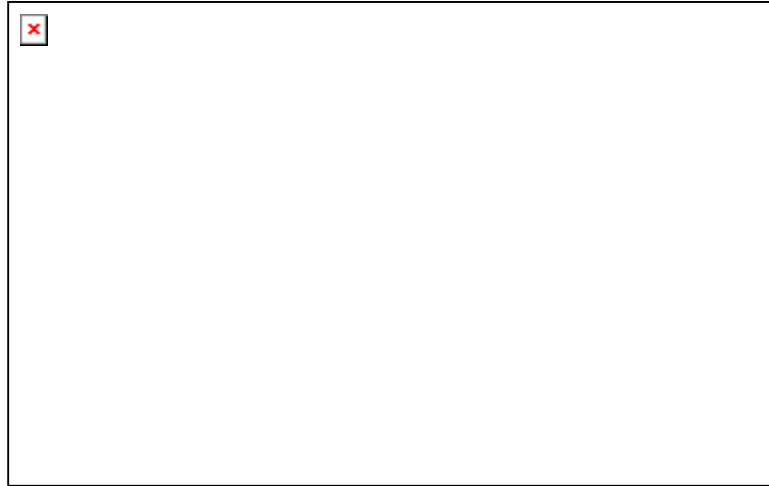
Utiliza complemento a 2 para los números negativos.

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	1
P	SSM	Data																			Pad			SDI	LABEL

**Figura 64 Ejemplo trama BNR ARINC429**

- Discrete Bits. Permite el uso del mensaje como transmisión de señales discretas. Por ejemplo ejemplo:

Bit	Function	1	0
1	Label 005	X	
2	Label 005		X
3	Label 005	X	
4	Label 005		X
5	Label 005		X
6	Label 005		X
7	Label 005		X
8	Label 005		X
9	SDI		
10	SDI		
11	PAD		X
12	PAD		X
13	Failure to clear serial data interrupt	Fail	Pass
14	ARINC received fail	Fail	Pass
15	PROM checksum fail	Fail	Pass
16	User RAM fail	Fail	Pass
17	NV RAM address fail	Fail	Pass
18	NV RAM bit fail	Fail	Pass
19	RTC fail	Fail	Pass



**Figura 65 Discrete Bits ARINC429**

- Maintenance and Acknowledgement. Se usa un intercambio de mensajes para resolver el estado de las comunicaciones o mandar información relevante de mantenimiento.
  - ISO Alphabet No. 5 y mantenimiento conocido como *AIM block transfer protocol*<sup>3</sup>.
- d) **Sign/Status Matriz (SSM).** Usa los bits 30 y 31 para enviar el modo de operación, condiciones de los equipos HW o validación de la información contenida en data.



**Figura 66 Matriz de Status ARINC429**

- e) **Paridad.** Se utiliza un esquema de paridad impar con “1” lógicos. Se realiza el recuento de número de “1” en el mensaje, si es par este campo valdrá “1”, si es impar valdrá “0”.

---

<sup>3</sup> Posteriormente se reemplaza con Williamsburg File Transfer Protocol.

### AIII.iii. AFDX

#### II. Conceptos AFDX

Es necesario explicar algunos conceptos que definan más claramente la tecnología AFDX [30], [31]. Se realizará una definición por separado aunque no cabe duda de que están fuertemente entrelazados.

##### a. Determinismo

Es el concepto principal que se pretende conseguir con AFDX. El objetivo es conseguir que las características de la red no sean estocásticas<sup>4</sup>. Es necesario tener controlado en todo momento el estado y capacidad de la red.

Se consigue gracias a la redundancia de la red, el concepto de tubería virtual o Virtual Link (VL) y el ancho de banda fijo para cada emisor (BAG).

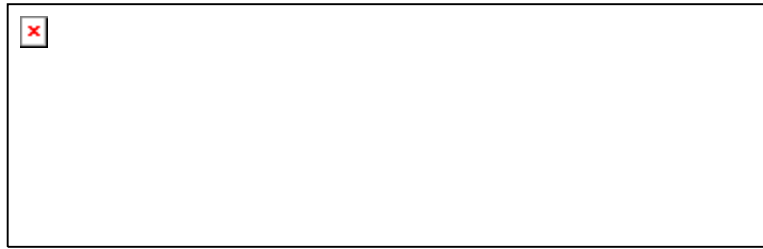
##### b. Redundancia AFDX

En aviónica no se puede permitir un fallo grave de las comunicaciones, por lo que para mejorar la red de transmisión de los mensajes se duplica completamente toda la red. Esto implica, que un fallo de un *switch* AFDX en una red no tiene un impacto directo en la red, ya que está duplicada.

En transmisión, todos los mensajes se duplican y envían por ambas redes. Para recepción se utiliza un algoritmo de *first valid wins*, el primer mensaje correcto en llegar será el que se emplee, y el otro se descarta. El servicio será transparente a la aplicación.

---

<sup>4</sup> Funcionamiento basado en el azar



**Figura 67 Redundancia AFDX**

En transmisión, la aplicación pasa los datos de envío a la *communications protocol stack*. Se emplean los protocolos UDP, IP para enlazar diferentes aplicaciones en un mismo VL y un control de número de secuencias para mejorar la recepción, y evitar tramas repetidas. En recepción se descartan aquellas tramas que ya se hayan recibido.

c. Virtual Link

El concepto de VL no es un concepto propio de AFDX. Es la capacidad de la red de crear una tubería virtual que mantendrá unas propiedades durante su uso. VL define un canal virtual que conecta un transmisor y uno o varios receptores.

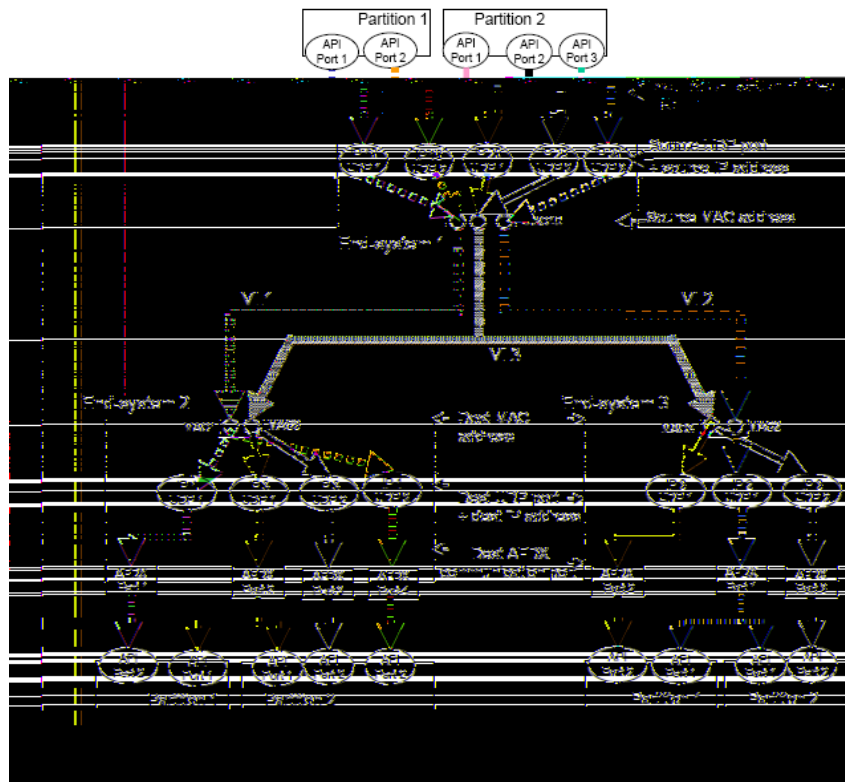


Figura 68 Virtual Link AFDX

Ya que cada *switch* tiene conexión directa con cada ES, se emplean las direcciones MAC para identificar la VL. En cada interfaz del *switch* sólo se tiene un equipo conectado, así que no es necesario identificar unívocamente, cada ES pertenecerá a una VL. Se emplean los dos últimos bytes de la dirección MAC como identificador.

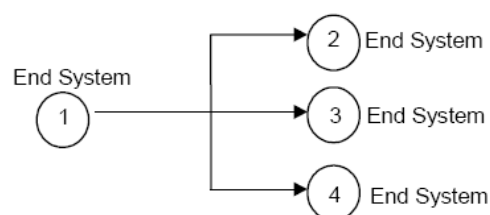


Figura 69 Virtual Link AFDX 2

Cada VL mantendrá un ancho de banda único lo que permitirá conseguir un retardo determinista:



El tiempo de latencia de un mensaje sin errores ni colisiones será:

$$L = T_s + T_m + T_r$$

$T_s$  => Tiempo de latencia de transmisión.

$T_m$  => Tiempo de latencia de la red.

$T_r$  => Tiempo de procesamiento de ES en recepción.

$$T_m = \frac{Length}{BW}$$

En este caso, se tiene un ancho de banda fijo gracias al VL creado, no hay colisiones gracias a una red *Switched Ethernet*, y además se establece un tamaño máximo de mensaje, por lo que se consigue definitivamente un retardo determinista.

El tiempo de retardo, al ser determinista, podrá ser calculado en cualquier momento. Aumentando el ancho de banda, se reduce el tiempo de retardo. En el caso de utilizar *Switched Ethernet*, se añade un tiempo de proceso del *switch*, que también está acotado, ya que tiene un búfer con un tamaño limitado.

Para conseguir el ancho de banda fijo, se utiliza un tiempo de guarda (**B**andwidth **A**llocation **G**AP **B**AG), que deberá respetar el emisor. Si se tiene una longitud máxima del mensaje, y un tiempo de guarda entre cada trama enviada, se establece un ancho de banda fijo:

$$BW = \frac{Length}{BAG}$$

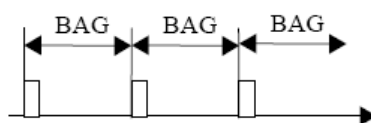


Figura 70 BAG

El control de flujo se realiza en transmisión y en *switching*. El transmisor dispone de un *scheduling* que ordenará el envío de acuerdo con el ancho de banda reservado, y el tamaño máximo permitido para los mensajes. El *switch* también controla los valores de BAG y tamaño máximo, descartando aquellos paquetes que no cumplan las especificaciones. En recepción no se controlará.

d. AFDX ports

Se definen dos tipos de puertos para garantizar dos tipos de servicios: *Queuing* y *Sampling*. Ambos están diseñados para soportar servicios ARINC 653.

**Sampling:**

- En este tipo de puerto, cada dato nuevo que llega sobrescribe al anterior dato almacenado.
- Sólo permite 1471bytes de datos.
- No se permite fragmentación IP.
- Mientras el mensaje está en el búfer todas las aplicaciones lo podrán leer.
- Hay información temporal de refresco de los datos.

**Queuing:**

- En este puerto, cada nueva trama que llega es considerada como nueva información y pasada a la siguiente capa.
- En este tipo de puertos se permite fragmentación IP.
- Cada puerto maneja hasta 8Kbytes de datos de aplicación.
- Si hay *overflow* se pasa el mensaje a la aplicación.

Hay otro tipo de puerto llamado SAP, que es un puerto *queuing* que incluye capacidad de mantenimiento e información de la red (SAP).

### III. Formato de Trama

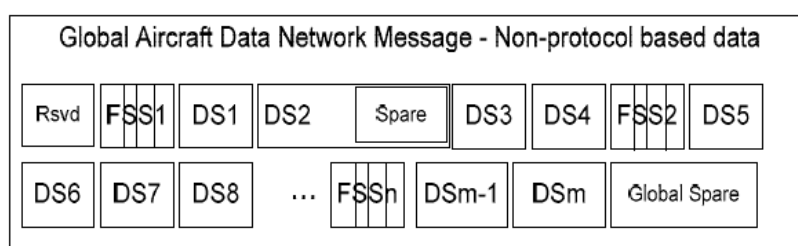
La trama de datos propia de AFDX tendrá 2 partes fundamentales:

- **Funcional Status Set (FSS):** Es un campo de 32 bits dividido en 4 grupos de 8 bits. Cada grupo establece el status y condición de cada campo DS. Los valores que puede tener cada campo de FSS son:

Condition	Definition
ND (no data)	No valid data in data set, this would include Fail Warn and other conditions where the contents are meaningless.
NO (normal operation)	Valid data, Normal operating conditions
FT (Functional test)	Equipment test conditions
NCD (no computed data)	Invalid data, equipment is in normal operating conditions but unable to compute reliable data.

**Figura 71 FSS AFDX**

- **Data Set (DS):** Es el campo que contiene los datos de la aplicación, y que puede estar compuesto por uno o más datos primitivos (*integer, float...*). Podrá tener a su vez un relleno (*spare*) para mantener el BW predefinido.



**Figura 72 Data Set AFDX**

La composición final del mensaje se muestra en las siguientes figuras. Los campos de datos y de control de los datos se acumulan hasta formar los datos finales de la trama Ethernet que se envía por la red.

La trama Ethernet es una trama clásica, a la que se añaden bytes de GAP para garantizar la separación entre tramas.

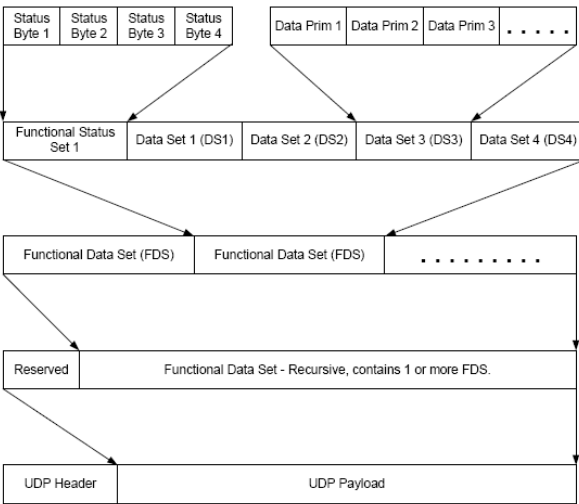


Figura 73 Intercalado de tramas AFDX

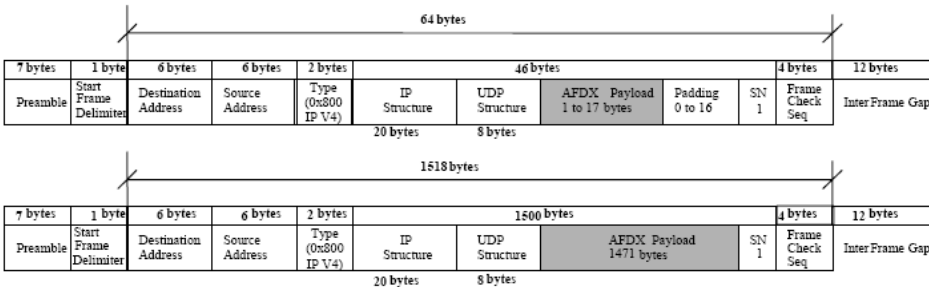


Figura 74 Trama Ethernet

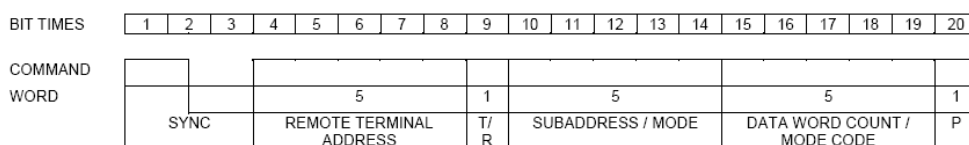
**AIII.iv. MIL-STD-1553**

La definición correcta del bus MIL-STD-1553 [33], [34], [35] exige una especificación completa de las características principales.

**II. Tipos de palabras**

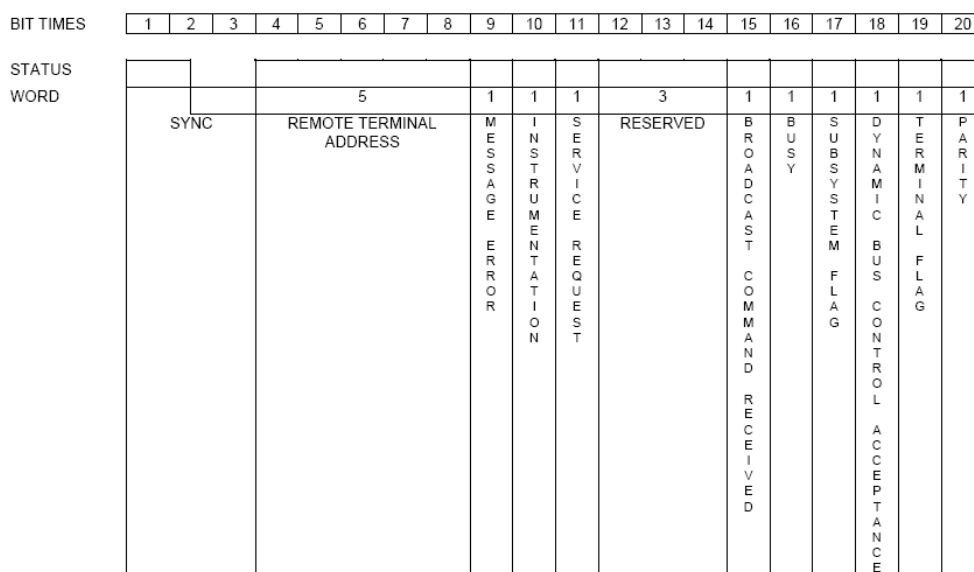
Se especifican tres tipos de palabras que se envían por el bus de datos:

- **Command Word:** Esta palabra sólo la envía el BC hacia un RT. Va acompañada de datos en caso de que se estén enviando, o será una petición de envío de datos.



**Figura 75 Command Word 1553**

- **Status Word:** Sólo es enviada por un RT indicando su estado o en su caso, algún error detectado en la información recibida, en el bus, o en su estado de error.



**Figura 76 Status Word 1553**

- **Data Word:** La puede enviar tanto BC como RT y contiene la información que se está enviando desde un equipo de aviónica a otro.

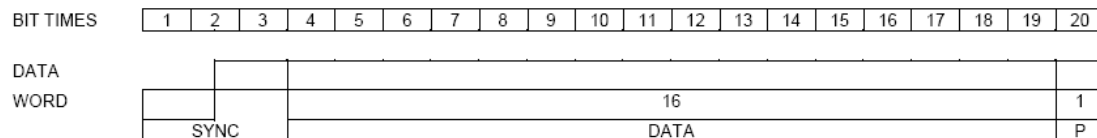


Figura 77 Data Word 1553

La información a través del bus de datos está codificada con Manchester II. “1” lógico corresponde con nivel alto durante la mitad del período y nivel bajo la otra mitad. “0” lógico será lo contrario, nivel bajo seguido de nivel alto. En esta codificación se permiten 2 palabras especiales, *Command* y *Status*. La estructura de ambas es la misma, pero se diferencian en que cada una se usa en un momento distinto.

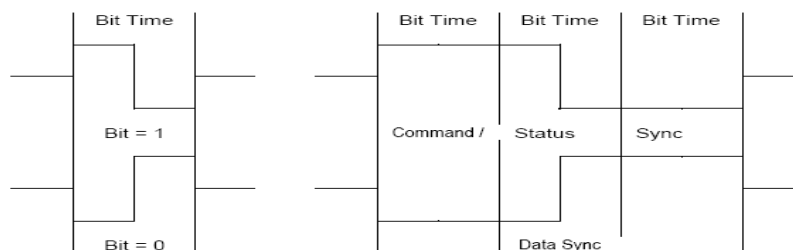


Figura 78 Codificación Manchester 1553

### III. Protocolo

Protocolo es la definición sobre cómo se usan los 3 tipos de palabras para transferir los datos entre los equipos. Se definen 10 tipos diferentes pero todos siguen la misma pauta:

- Todos los mensajes comienzan con un comando desde la BC.
- Hay un tiempo mínimo de separación entre mensajes.
- Si un RT tiene que responder a un comando, la respuesta comenzará con un *Status Word*.
- No debe haber tiempo de separación entre palabras de datos.

Hay 2 tipos de tiempos de guarda. Tiempo de respuesta de RT, es el tiempo entre el que debe responder el equipo (4-12 microsec.). Gap entre mensajes es el tiempo de guarda entre mensajes, y debe ser, como mínimo, 4microsec. Ambos tiempos se miden entre el cruce por cero del bit de paridad y el cruce por cero del bit de sincronismo.

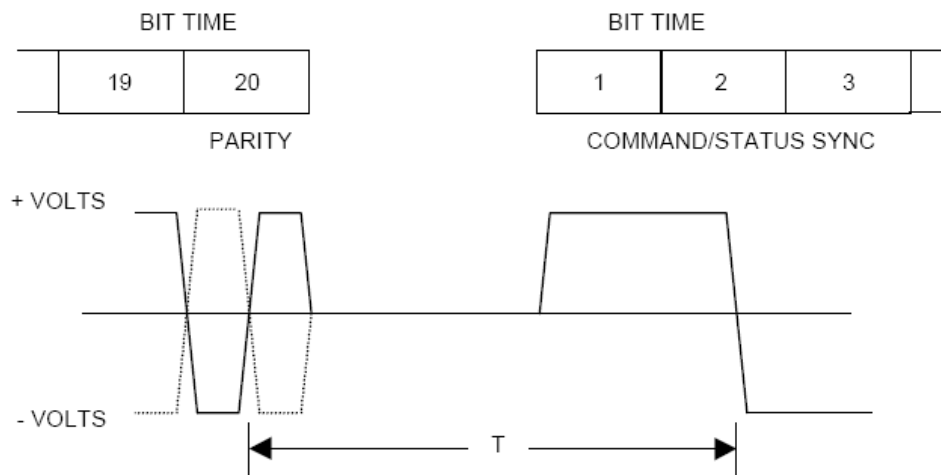


Figura 79 Timing 1553

- BC-RT: BC envía datos a RT. El bit T/R se pone a 0 indicando que va a recibir datos. RT recibe el comando y los datos, y si todo va bien responde con la palabra de status.



Figura 80 Envío BC-RT 1553

- RT-BC: BC envía un comando hacía el RT correspondiente indicando en el bit T/R a 1 que debe enviar datos. Cuando todo es adecuado se envía la palabra de *status* y las palabras de datos correspondientes.

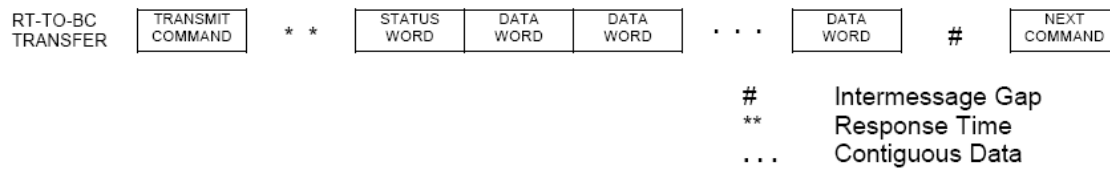


Figura 81 Envío RT-BC 1553

- RT-RT: BC es el que dirige el proceso. Primero envía un comando a la RT que va a recibir y después otro comando a la que debe enviar especificando los valores correspondientes. Después la RT debe enviar *status word* y datos, y la que recibe envía *status*.

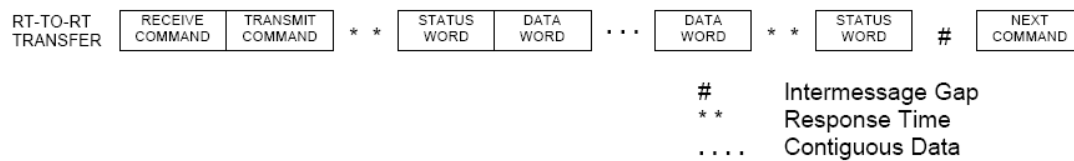


Figura 82 Envío RT-RT 1553

- Modo comando sin datos: Se envía un comando al equipo correspondiente para que realice ciertas actividades. Por ejemplo, se le puede enviar un comando de *restart*.

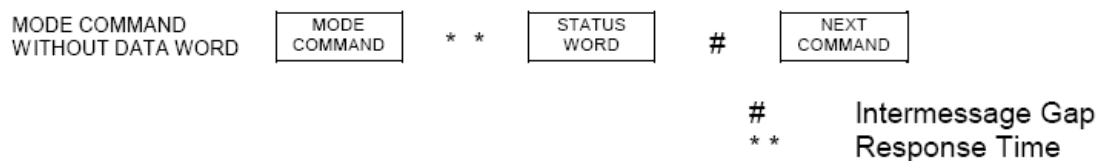


Figura 83 Modo comando sin datos 1553

- Modo comando con datos (transmitir): Algunos comandos requieren de cierta información adicional además de la palabra de status. Este es el caso por ejemplo de un *Built-In test*.



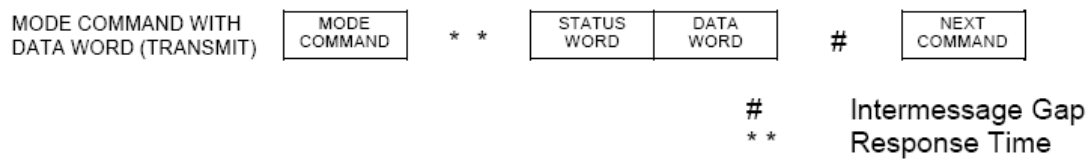


Figura 84 Modo comando con datos 1553

- Modo comando con datos (recibir): Otros tipos de comandos pueden requerir información adicional, como puede ser el comando de *shutdown*.



Figura 85 Modo comando con datos rx 1553

- Broadcast: BC tiene capacidad de enviar algunos mensajes en modo broadcast, hacia todos los RT's. La dirección del RT en este caso será la 31 (11111).
  - BC-RT
  - RT-RT
  - Modo comando con datos (recibir)
  - Modo comando sin datos

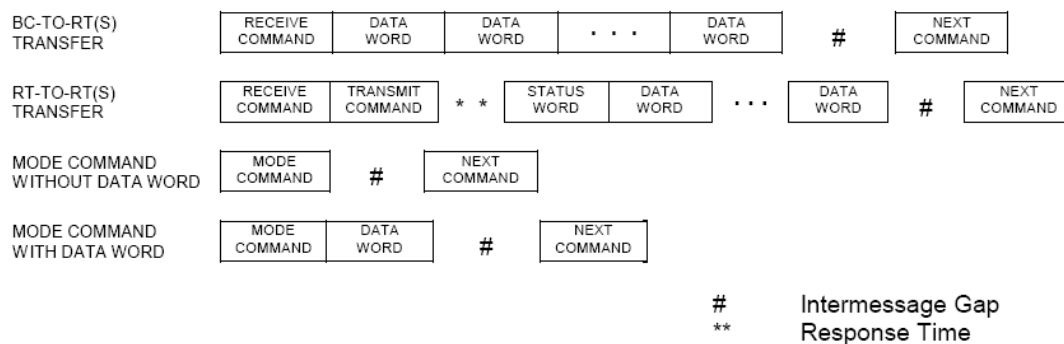
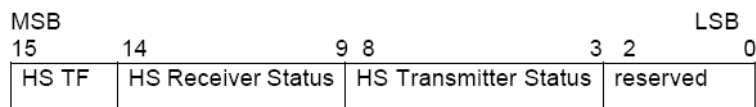


Figura 86 Otros modos 1553



Palabra de estado en el bus LS enviada hacia algún terminal RT para que especifique su estado por HS.

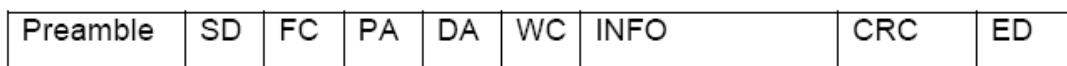


- HS TF: HS Terminal Flag ( optional )
- HS RX Status
  - Bit 14 :HS Message Frame Error
  - Bit 13 :HS Receiver Active
  - Bit 12 :HS Receiver not ready ( optional)
  - Bits 9...11: reserved (set to 0)
- HS TX Status
  - Bit 3 :HS Transmitter active
  - Bit 4: HS Transmitter not readies (optional)
  - Bits 5...8: reserved (set to 0)

**Figura 88 HS Status Word 3910**

## • HS Message Word

Las longitudes mínimas y máximas del mensaje son 624bits y 65648bits.



**Figura 89 HS Message Word 3910**

Preámbulo: 40 bits para control de ganancia, sincronización e inicio de trama.

Start Delimiter (SD): 8 bits únicos que indican el comienzo de la trama HS.

Headers: **Frame Control (FC):** patrón único.

**Physical Address (PA):** Dirección del emisor RT.

**Destination Address (DA):** Dirección receptor RT.

**Word Count (WC):** Define las palabras de datos en info.

**Info:** Datos entre 32 y 4096 palabras de 16bits. Debe ser múltiplo de 32.

**CRC:** Control de corrección de datos.

**End Delimiter (ED):** Palabra única de fin de trama.

### AIII.vi. CAN

#### I. Protocolo

Para el control de acceso [37] se emplea CSMA/CD con prioridad de mensajes. La prioridad de cada equipo se establece en el inicio de la conexión y se resuelve por decisión. Se emplea *Bit-Wise* no destructivo:

1. Se envía la prioridad del mensaje.
2. Cuando uno de los terminales mande un bit dominante y otro terminal un bit recesivo, este último perderá la contienda.

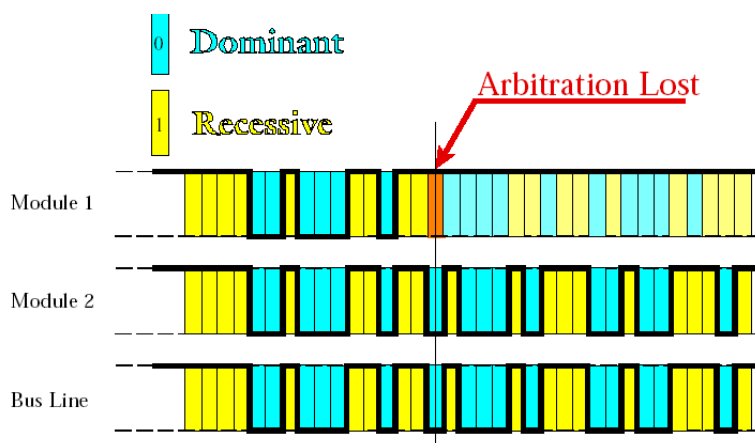


Figura 90 Bit-Wise Arbitration CAN

Todos los nodos del bus reciben el mismo mensaje, independientemente del número que haya conectado. Y cualquiera puede enviar en el momento que el canal esté libre.

Sin embargo, la especificación bus no define capas superiores de protocolos, por lo que para más precisión en la comunicación de deberán especificar el resto de niveles de comunicación. Hay varios protocolos de aplicación ya definidos para el control del bus CAN como son CAL, CANOpen, SDS...

## II. Tipos de Mensajes

Hay cuatro tipos de mensajes definidos en las especificaciones CAN.

- **Data Frame:** Envía información desde el transmisor hacia uno o varios receptores.

1	12/32 bits	6 bits	0 – 8 bytes	16 bits	2 bits	7 bits	$\geq 0$
SO	Arbitration field	Control field	Data field	CRC field	ACK field	EOF	IFS

**Figura 91 Data Frame CAN**

Existen siete campos de trama definidos:

**Start of Frame (SOF):** Cambio de bit que indica el inicio de la trama de datos.

**Arbitration Priority and Identification Field:** Versión estándar (12bit) o versión extendida (32bit). Dos posibilidades de actuación, como identificador (11 o 29bits) o como resolución de conflicto (12 o 32bits).

Este campo marca la prioridad del mensaje en caso de colisión. Los controladores CAN pueden utilizar este campo para filtrar los mensajes recibidos por el bus.

Dentro de este campo se tiene un bit al final RTR, que indica si la trama es de datos o de control remoto. En la trama extendida hay otro bit **Substitute Remote Request (SRR)**.

	Arbitration field		Control field			Data Field	CRC Field	Ack field	
1	11 bits	1	1	1	4 bits	0..64 bits	16 bits	2 bits	7 bits
S O F	11 bits identifier	R T R	I D E	R O	DLC	0 – 8 bytes	CRC	Ack	EOF

**Figura 92 Data Frame 1 CAN**

Star of frame	Arbitration field				Control field				Data Field	CRC Field	Ack field			End of frame
1	11 bits	1	1	18 bits	1	1	1	4	0 bit	15	1	1	1	7 bits
S O F	11 bit identifier	S R R	I D E	18 bit identifier	R T R	R 1	R 0	DLC	N/A	S E Q	A C K	S L O T	D E L	E O F

**Figura 93 Data Frame 2 CAN**

**Control:** La función principal del campo es el **Data Length Code (DLC)** que puede tener valores de 0..8. Los otros 2 bits están reservados para indicar trama extendida. DLC indica el número de bytes que existen en el campo de datos.

**Data:** Este campo puede tener entre 0 y 8 bytes de datos indicados según el DLC. Cada byte tiene siempre 8 bits y se transmite el MSB primero. Algunos equipos extienden el identificador de mensaje hasta los datos.

**CRC:** Control de redundancia cíclica adaptado a la longitud del mensaje. Es sólo uno de los mecanismos de control de errores de CAN.

**ACK:** Campo de Acknowledgment de recepción. Con esto se asegura que al menos un receptor lo ha recibido adecuadamente. El transmisor pone este bit a “1” y los que lo reciben bien contestan con el campo a “0”.

**EOF:** Bits de fin de trama.

- **Remote Frame:** En configuración Maestro/Esclavo o Multimaestro, un maestro pide información a un transmisor.

Esta trama es la misma que la de datos con dos diferencias. El bit RTR está marcado a “1” indicando que es una trama de control remoto, y no contiene campo de datos.

En este caso, el DLC indica el número de bytes de envío que se están pidiendo.

Start Of frame	Arbitration field				Control field				Data Field	CRC Field		Ack field		End of frame
1	11 bits	1	1	18 bits	1	1	1	4	0..64	15	1	1	1	7 bits
S O F	11 bit identifier	S R R	I D E	18 bit identifier	R T R	R 1	R 0	DCL	0-8 Bytes	S E Q	A C K	S L O T	D E L	E O F

**Figura 94 Remote Frame 3 CAN**

- **Error Frame:** Es transmitido por cualquier transmisor cuando detecta un error en el bus de datos.

Esta trama consiste en un flag de error de 6 bits. Tiene todos los bits a “0” en caso de que sea enviado como activo, y estarán todos ellos a “1” en caso de que sea pasivo.

El campo Error Delimiter son 8 bits a “1” en el que otros equipos pueden enviar sus errores cuando detectan el primer flag de error.

Dataframe	Error Frame		IFS or Overload frame
	Error Flag 6 bits of the same value	Error Delimiter 8 recessive	
0-64 bits	6- 12 bits	8 bits recessive	3 bits

Figura 95 Error Frame CAN

- **Overload Frame:** Tiempo extra de retardo entre dos mensajes.

Se envía por un terminal cuando está demasiado ocupado.

End of frame or Error delimiter or Overload delimiter	Overload frame		IFS or Overload frame
	Overload Flag	Overload Delimiter	
	6 dominants bits	8 recessive bits	

Figura 96 Overload Frame CAN

### III. Error Handling

Aprovechando que todas las estaciones reciben todos los mensajes se puede hacer un control de errores distribuido. El transmisor escucha mientras emite, evitando colisiones y además, cada terminal mantendrá dos contadores de errores, *Transmit Error Counter* y *Recieve Error Counter*. En la inicialización, el terminal entra en el estado *Error Active*. Cuando alguno de los dos contadores de errores supera el valor 127 entra en modo *Error Passive*, y cuando supera el valor 255 entra en *Bus Off State*.

En el valor activo envía *error flags* activos cuando detecte un error. En pasivo envía *error flags* pasivos cuando detecte error. En *bus off state* no envia nada por el bus, ni datos ni errores.

Además del uso de ACK, CRC, *Frame Check* y *Bit Monitoring*, se hace uso del *Bit Stuffing*, ya que debido a la codificación NRZ empleada puede haber problemas de sincronismo. Cuando se detectan 5 bits consecutivos de



idéntico valor se introduce un bit de valor complementario. Por lo tanto se pueden detectar 5 tipos de errores en el bus de datos.

Cuando se detecta un error de CRC un *error flag* se envía justo después del ACK. Si se detecta uno de los otros errores se comienza un *error flag* justo después de la detección.

#### IV. Capa física

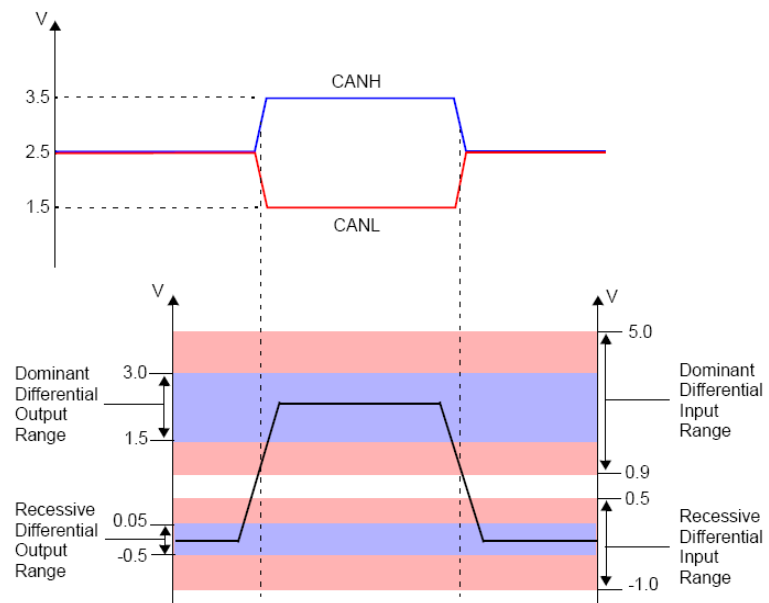
En la capa física se define el tipo de cableado a emplear en el bus. En concreto se permiten:

- Twisted Pair Bus
- Single Wire Bus
- Optical Passive Star
- Optical Active Star

Bus length	Bit rate kbit/s	Nom bit time $\mu$ s	Comments
25 m	1000	1	In-out delays 210 ns
50 m	800	1.25	
100 m	500	2	In-out delays 300 ns (incl 2x40 ns optocoupl)
250 m	250	4	
500 m	125	8	In-out delays 450 ns (incl 2x100 ns optocoupl)
650 m	100	10	
1 km	50	20	Propagation delay 5 ns/m 10 ns delay = 2m
2.5 km	20	50	
5 km	10	100	

**Figura 97 Características CAN**

En el siguiente cuadro se representan los valores de voltajes aceptados en el protocolo CAN.



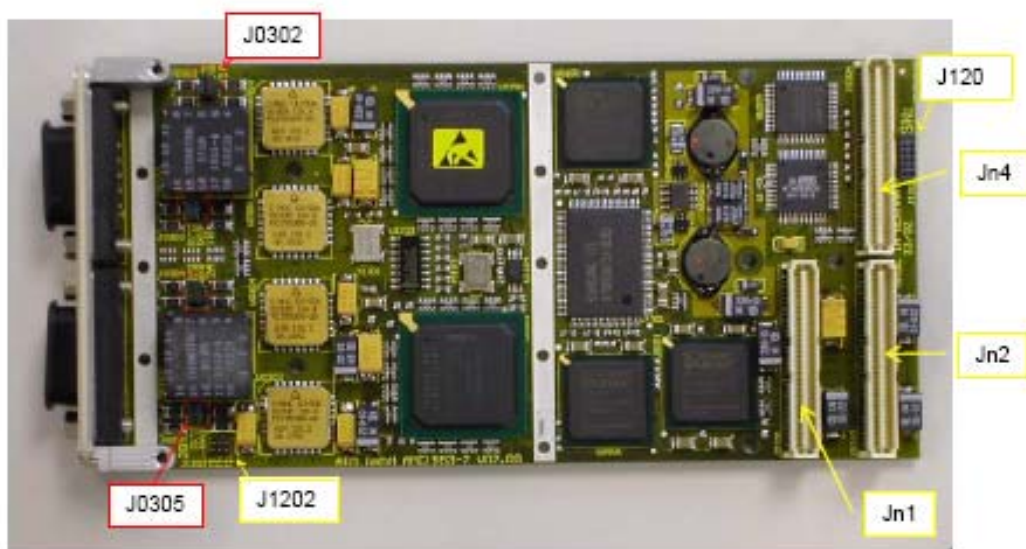
**Figura 98 Capa Física CAN**

#### **Anexo IV. Tarjetas de comunicaciones empleadas**

Estos son sólo algunos ejemplos para describir mejor y con más claridad el concepto de tarjetas de comunicaciones. Hay gran variedad de tarjetas con diferentes capacidades y características.

##### **AIV.i. AMC 1553**

La tarjeta AMC1553 [39] de AIM está rediseñada para analizar, simular, monitorizar y probar sistemas de aviónica basados en el bus de datos MIL-STD-1553. El interfaz de tarjeta implementa IN/OUT para cada canal e IRIG IN/OUT.



**Figura 99 AMC1553 [39]**

Esta tarjeta tiene un modulo PMC diseñado para ser montado sobre una tarjeta de adaptación a cualquier estándar PCI, VME o CPCI, o un PC integrado con puertos PMC.

La arquitectura de la tarjeta emplea dos procesadores, uno para cada canal de 1553 de los disponibles. Dispone de una controladora PCI y un controlador de memoria realizados en FPGA. Esta FPGA soporta a la vez la

interfaz de la aplicación y tareas de software lanzadas en el PC servidor, y además es capaz de realizar la transmisión y recepción de datos. Para conseguir las necesidades de tiempo real de los buses de aviónica provee de un procesador RISC de 32 bit (BIP) para cada BIU 1553.

La estructura de la tarjeta se define a continuación. AMC1553-1 implementa sólo la sección BIU 1, mientras que AMC1553-2 implementa BIU 1 y BIU 2.

- PCI y FPGA.
- RAM.
- Sección BIU.
- Interfaz físico de entrada y salida con 2 canales 1553.
- IRIG Timecode Processor
- Alimentación (Power Supply)
- Soporte JTAG

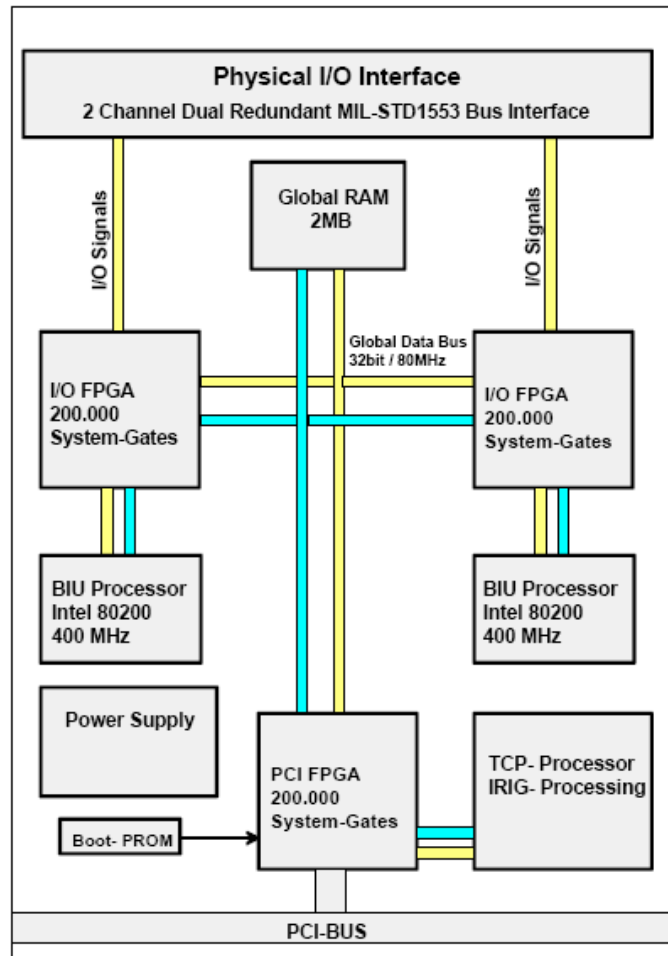


Figura 100 Esquema AMC1553 [39]

#### AIV.ii. AMC 429

AMC429 [40] es una tarjeta de AIM diseñada para simular, analizar, monitorizar y probar sistemas de aviónica basados en el bus ARINC429.

AIM provee un modulo PMC que permite el montaje sobre tarjetas de adaptación a PCI, VME o CPCI, además de tener la capacidad de conectarse a un servidor integrado con puertos PMC.

Esta tarjeta tiene 4,8 o 16 canales ARINC429 totalmente configurables. Las versiones de 4 y 8 son una modificación de la versión de 16 canales. Cada canal puede ser configurado como emisor o transmisor.

Una nueva versión con 32 canales está disponible, como ampliación de la versión de 16 canales. Los 16 canales más bajos están diseñados para ser programados como transmisor o receptor, y los 16 canales superiores transmiten y reciben dependiendo de las necesidades puntuales.

Debido a la falta de pines de entrada y salida, las entradas y salidas (IO) IRIG en la tarjeta, están disponibles en los IO frontales. En transmisión, la tarjeta actúa autónomamente como un bus de tráfico, permitiendo múltiples modos de transmisión e introducción de errores. La programación de errores es individual para cada canal y cada mensaje.

Para algunos modos especiales de transmisión se podrá emplear el bit de paridad como datos. Los tiempos de subida y bajada de las señales del bus son también programables por software para cada canal.

En los canales de recepción, se dispone de funciones de análisis y monitorización con detección de errores, muestreo y filtrado.

La arquitectura de la tarjeta garantiza la separación real de funciones en ejecución, sin afectar unas a otras. Las diferentes velocidades de transmisión se pueden programar desde 90 a 120Kbps en bus rápido y 11.5 a 16Kbps en líneas de baja velocidad.

La estructura de la tarjeta se muestra a continuación:

- ASP
- PCI
- Sección BIU
- RAM
- Time Code
- Interfaz de bus físico

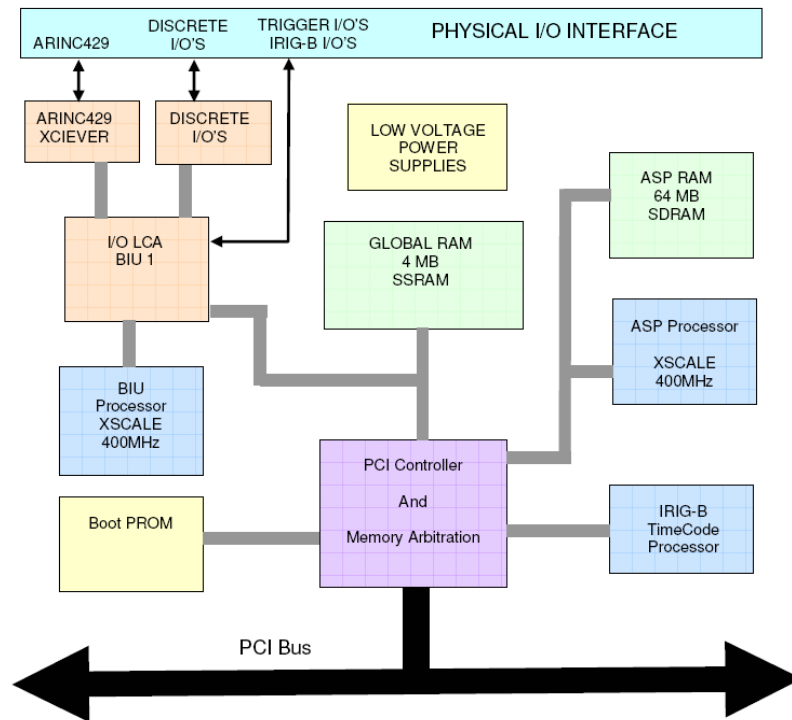


Figura 101 Esquema API429 [40]

#### AIV.iii. AVI 3910

AVI3910 [41] es una nueva tarjeta perteneciente a los módulos de VME-Bus, empleada para analizar, monitorizar y probar sistemas de aviónica basados en el bus STANAG3910.

Se emplea esta tarjeta para simulación, monitorización y estimulación de errores en el bus. Con esta tarjeta se dispone de dos canales 1553 y dos canales 3910 sobre un mismo *slot* VME. El enlace de FO de 3910 está implementado en otro módulo, VOX3910, que proporciona la capacidad de conexión por FO necesaria en 3910.

Con la arquitectura HW disponible, existen suficientes recursos (procesamiento y memoria) para garantizar que todas las funciones pueden utilizarse simultáneamente. Al tener capacidad de procesamiento dentro de la tarjeta y gran memoria DRAM, puede realizar operaciones de forma aislada sin necesitar recursos del servidor PC. Tiene tres procesadores:

- 64bit RISC (ASP) para dar soporte a las aplicaciones y el *driver* SW, y aumentar la capacidad de los módulos de la tarjeta.
- 32bit RISC (BIP) en cada BIU para cumplir los requisitos de tiempo real.

El decodificador de tiempos IRIG B está implementado para cumplir los requerimientos de *multi-channel synchronization*.

La estructura se presenta a continuación:

- ASP
- BIU
- RAM Global
- Time Code
- Bus físico



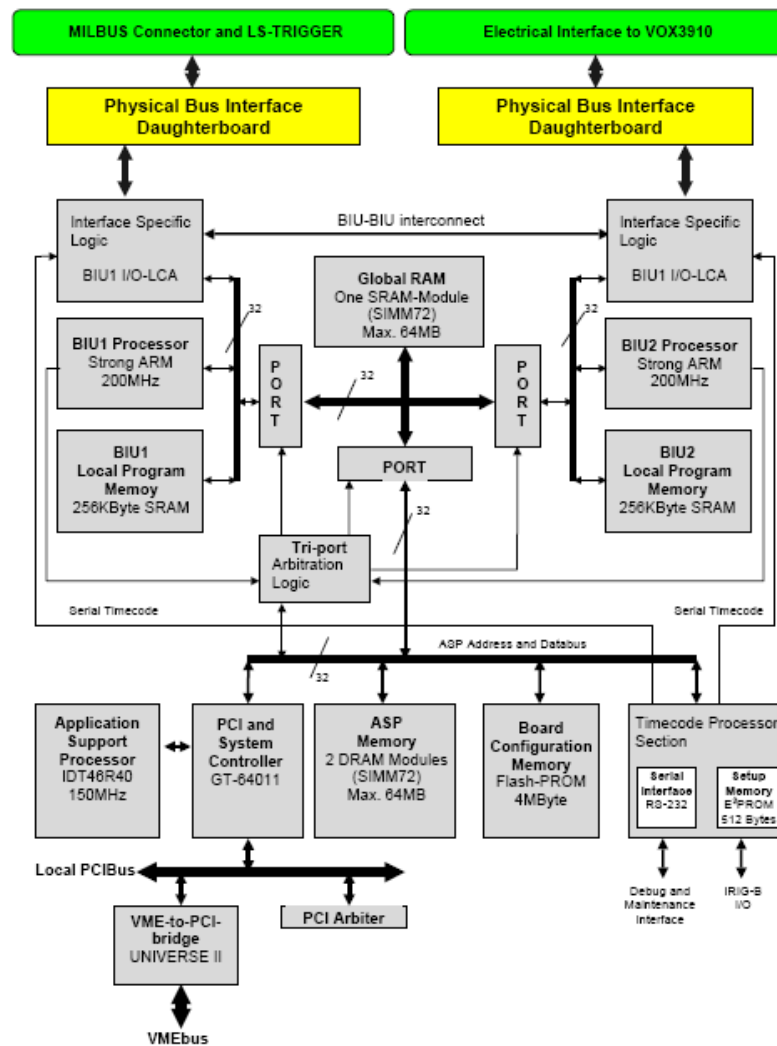


Figura 102 Esquema AVI3910 [41]

#### AIV.iv. PCI-CAN/XS2

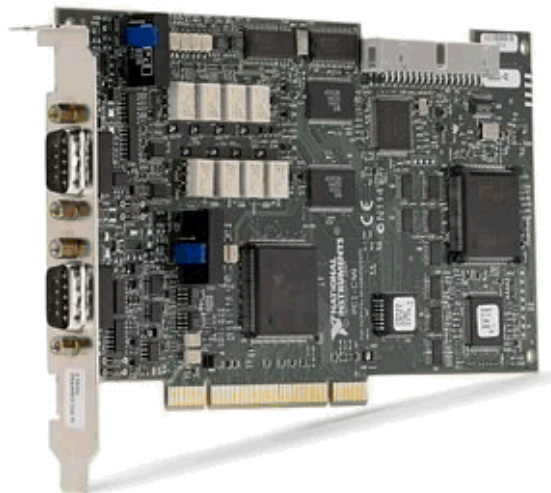
La tarjeta PCI-CAN/XS2 [42] de National Instruments (NI) es un interfaz CAN de dos puertos seleccionables por transceptor diseñada para comunicar con dispositivos CAN de alta velocidad, de baja velocidad o de un solo cable.

El transceptor adecuado para cada puerto es seleccionado por SW a través de una aplicación ofrecida por NI. Los módulos NI CAN para PCI utilizan el Controlador Philips SJA1000 CAN para funcionalidad avanzada como escuchar únicamente, recepción personal (echo) y modos avanzados de filtrado y nuevos transceptores para modo *sleep/wakeup*.

Cada interfaz CAN se vende con el SW de dispositivo NI-CAN. Se pueden sincronizar tarjetas CAN, DAQ, de visión y de movimiento usando el bus RTSI. Todas las interfaces CAN de NI están diseñadas para cumplir con los requerimientos físicos y eléctricos para redes basadas en CAN.

La tarjeta lleva integrado un procesador Intel80386EX, que permite el procesamiento de las comunicaciones directamente en la tarjeta, evitando que se pierdan tramas debido a saturación del servidor PC.

También tiene capacidad propia de temporización y muestreo de las señales en el bus PXI, PCI y PCMCIA/CAN gracias a que estas tareas están implementadas en HW. El servidor sólo interviene para coger las señales cuando están ya adquiridas.



**Figura 103 PCI/CAN XS2 [42]**

## Anexo V. Redes de conexión en matrices de conmutación

### AV.i. Introducción

Se presenta una definición de las partes más importantes, así como clasificación y características de las matrices de conmutación.

Las redes de conexión [1] de las matrices de conmutación están estructuradas dependiendo de las necesidades de cada cliente. Las características principales disponibles en el mercado se definen a continuación.

### AV.ii. Clasificación por puntos de cruce

Esta red de conexión, es la que se encargará de establecer el contacto físico entre los canales deseados, dependiendo de las posibilidades. Estas redes pueden tener diferentes configuraciones:

$N_x \equiv$  Número de puntos de cruce

- N entradas con M salidas

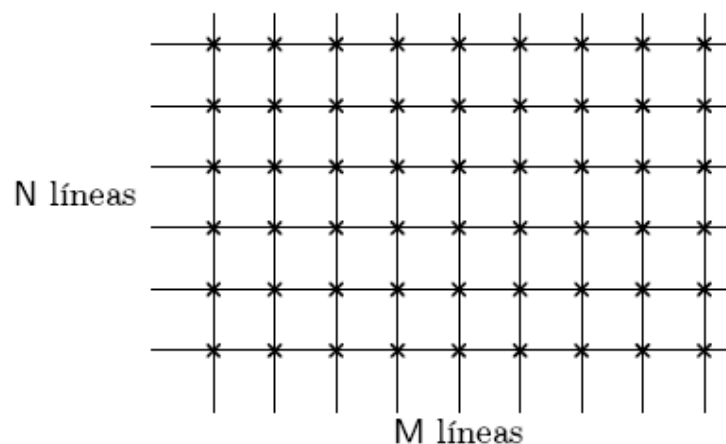


Figura 104 Puntos de cruce NxM [1]

$$N_x = N \times M \text{ (Bidireccionales)}$$

Cada entrada de la red puede conectarse con cualquiera o cualesquiera de las salidas. No hay restricción física para conectar varias entradas a una salida o varias salidas a una entrada, las restricciones serán lógicas, de potencia, o de razonamiento.

- N líneas

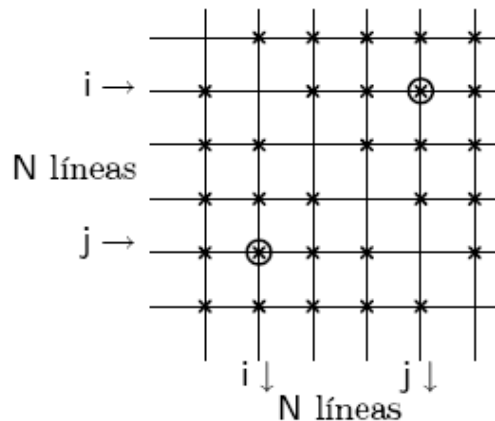


Figura 105 Puntos de cruce de N [1]

$$N_x = N^2 - N \text{ (Monodireccionales)}$$

Tenemos la posibilidad de conectar todos los puntos de entrada entre ellos, excepto obviamente, uno consigo mismo. Este será el modelo para arquitecturas en las que entradas y salidas no son algo claramente definido, por lo que necesitamos cualquier punto de la red con cualquier punto.

- N líneas plegada

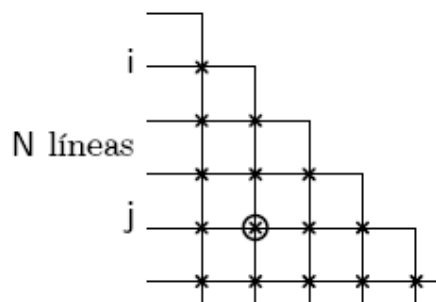


Figura 106 Puntos de N plegada [1]

$$N_x = \frac{N^2 - N}{2} \text{ (Bidireccionales)}$$

Al igual que antes, tenemos N líneas (entrada, salida o e/s) que podrán ser conectas cualquiera con cualquiera.

#### **AV.iii. Infrautilización**

Aunque estas matrices nos ofrecen capacidad de conexión completa entre cualquier par entrada/salida, es posible que no necesitemos todos estos recursos, por lo que habríamos gastado nuestros recursos económicos en tener demasiados puntos de cruce. Ante esta situación existen diversas opciones que no estudiaremos, puesto que nuestras necesidades si que son completas.

Aunque se podría realizar un estudio sobre el uso real de los equipos y sus conexiones, preferimos tener capacidad completa de uso y futura ampliación, ya que nuestro objetivo es precisamente tener toda la capacidad posible.

#### **AV.iv. Clasificación por tecnología de conmutación**

Existen tecnologías de conmutación mecánicas y electrónicas. Actualmente las tecnologías electrónicas se están imponiendo sobre las mecánicas, gracias a su menor tamaño y su mayor velocidad y duración. A pesar de ello, cada fabricante tomará sus elecciones y dispondremos de unas características distintas.

Las tres tecnologías más empleadas actualmente de conmutación son las siguientes:

##### **a. Solid State Rele (SSR)**

Es una tecnología [56] de conmutación optoelectrónica. Un relé de estado sólido se construye utilizando un MOSFET foto-sensible (**M**etal-**O**xide **S**emiconductor **F**ield **E**ffect **T**ransistor) controlado mediante la luz emitida por un LED, accionando el MOSFET foto-sensible, creando el canal entre fuente y drenador, y permitiendo el paso de corriente.

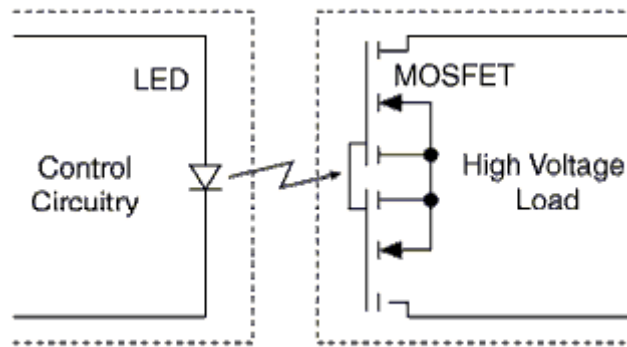


Figura 107 Esquema de SSR [56]

Son muy útiles para aplicaciones de alta tensión, ya que se separa el circuito del control del circuito de acción. Además, al no ser mecánicos sino electrónicos, son de una mayor duración.

Sin embargo, el tiempo de respuesta depende del tiempo de encendido y apagado del LED.

b. Field Effect Transistor (FET)

Esta tecnología es también electrónica. Se utilizan una serie de transistores CMOS para conectar o desconectar el circuito.

Al igual que los SSR, no es un dispositivo mecánico, por lo que tiene mayor duración. Sin embargo, el circuito de control actúa directamente sobre el transistor, por lo que son empleados principalmente en aplicaciones de baja potencia.

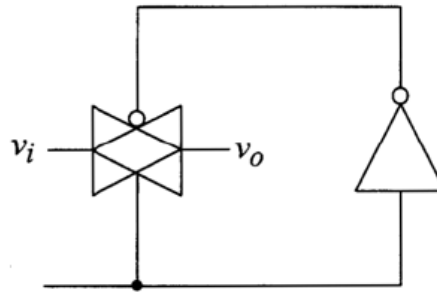


Figura 108 Conmutador CMOS

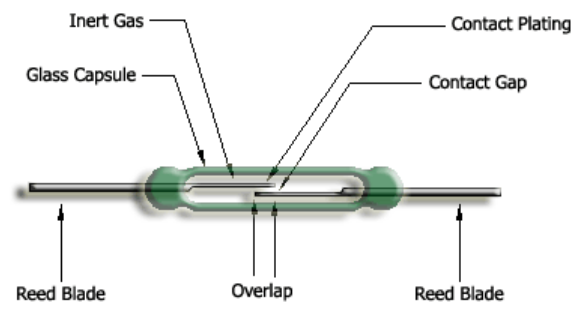
A pesar de que ambos sistemas son más rápidos y más duraderos, tienen una resistencia demasiado elevada como para emplearlos en sistemas de pruebas automáticas. Actualmente esto se está mejorando, y se consiguen resistencias equivalentes a los relés electromecánicos, con un coste mucho menor.

c. Reed

Este relé [57] es de tipo electromecánico, por lo que su vida útil es menor. Está compuesto por dos lengüetas (en inglés: *reed*) que se conectan cerrando el circuito a través de la activación de una fuerza magnética a la que se acerca el relé, generalmente una bobina que rodea el relé.

Se emplean en entornos explosivos o peligrosos, ya que la chispa producida al realizarse el contacto, queda retenida dentro del encapsulado.

Tienen una resistencia muy baja, mejorada además por metales como tungsteno o rodio colocados en el punto de contacto entre ambas lengüetas.



**Figura 109 Relé Reed [57]**



## Anexo VI. Proceso de desarrollo de bancos de integración

### AVI.i. Introducción

El proceso de desarrollo de bancos [58] y las actividades de soporte, se organizan en una lista de procesos que se relacionan entre sí. Cada proceso se caracteriza por:

- Un evento responsable de la ejecución del proceso.
- Una secuencia de actividades.
- Un conjunto de productos, o salidas finales.
- Una condición de fin.

Cada proceso puede realizarse para mejorar las capacidades de un banco específico o para mejorar las capacidades del Grupo de Desarrollo de Bancos (**B**ench **D**evelopment **O**rganization BDO).

El proceso de construcción de un banco se corresponde con la siguiente figura:

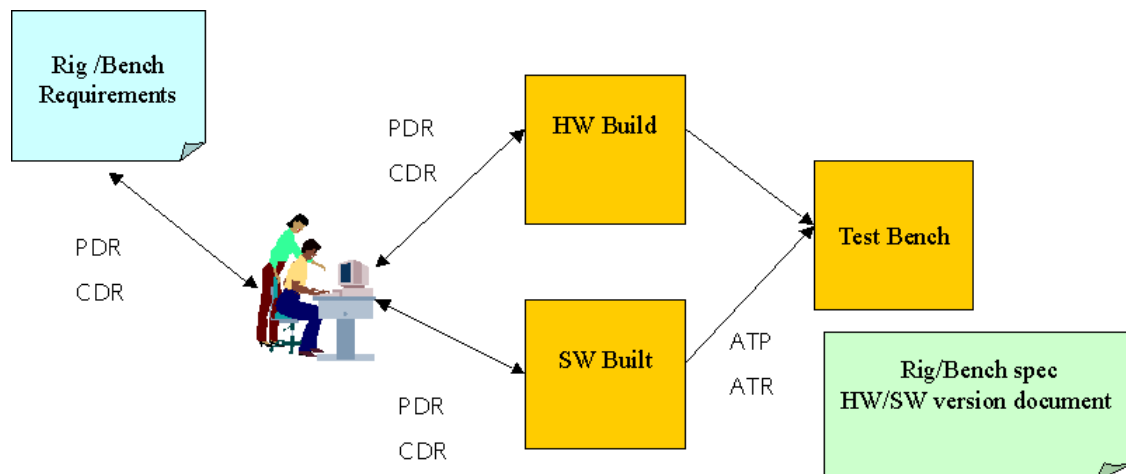


Figura 110 Proceso de construcción de banco [58]

### AVI.ii. Project Management

Este es el proceso principal. A partir de este proceso se llamará a otros procesos obteniendo salidas globales o entradas para otros procesos.

El evento responsable es una petición del *Engineering Programs Organization*.

La condición de fin del proceso es la aceptación del banco por parte del cliente.

Actividades:

I. Preparar propuesta

En caso de que la necesidad requerida por el *Engineering Programs Organization* no esté ya resuelta, se deberá preparar un proyecto. Este proyecto debe incluir:

- Especificación de costes
- Definición de tareas y riesgos

II. Desarrollo de plan y entrega

Cuando el *chief engineer* del área referente al banco en desarrollo aprueba la realización del proyecto, se realiza la planificación de actividades.

El objetivo de esta actividad es obtener:

- **Bench Quality Plan** (BQP), incluyendo también la nueva programación de tareas y riesgos.
- **Bench Investment Proposal** (BIP).

El BQP se realiza en paralelo junto con el análisis de requerimientos del banco, y se debe actualizar durante el diseño y construcción del banco, si es necesario.

El BQP debe ser aprobado por el responsable del departamento de *Test System* y por la *Quality Assurance Organization*.

En el BIP se definen las necesidades económicas del proyecto, así como las necesidades de inversiones en instrumentación o licencias. Este debe ser aprobado por el *Program Organization*.

## II. Dirigir el desarrollo

El jefe de proyecto debe:

- Activar el diseño e implementación cuando los recursos necesarios estén disponibles.
- Organizar el proceso de verificación, concertando citas periódicas con el cliente.
- Decidir sobre las discrepancias en el proceso de verificación.
- Iniciar el proceso de dirección de configuración para controlar la evolución de los datos ya verificados, y los elementos ya construidos.
- Lanzar el proceso de documentación cada vez que haya nuevos documentos que registrar.
- Comenzar el proceso de validación cuando los componentes del banco estén disponibles.
- Activar el proceso de dirección de configuración para registrar la nueva entrega.

El resultado de este proceso son informes mensuales que resumen:

- Actividades terminadas.
- Actualización del programa de tareas.
- Actualización de los riesgos.

### III. Entregar el banco y activar el soporte

El banco se entregará al cliente y se comenzarán las actividades de soporte para éste.

#### AVI.iii. Análisis de requerimientos

Los requerimientos del banco son generados por el cliente final. Normalmente el cliente será un grupo del interior de la empresa, aunque excepcionalmente puede ser un grupo exterior a ésta.

El evento responsable de la ejecución de este proceso es la necesidad de tener unos requerimientos fijos por parte del cliente.

La condición de fin es la validación del producto final por parte del cliente.

Actividades:

#### I. Revisar los requerimientos recibidos

- Información relacionada con la instalación de equipos y avión.
- Simulaciones necesarias.
- Descripción de señales e información del avión. Interface Control Data (ICD's)
- Necesidades de monitorización y almacenamiento de datos.
- Instrumentación específica
- Funciones especiales no desarrolladas ya por el grupo.
- Escenarios de test necesarios para pruebas automáticas.
- Necesidades especiales

La salida de esta actividad será el conjunto de dudas a realizar al cliente, como parte del bucle de realimentación.

## II. Formalizar requerimientos recibidos

Normalmente los requerimientos no llegan uniformemente, si no que por ejemplo, las necesidades HW llegan antes que las necesidades SW. Si se considera que los requerimientos ya definidos y corregidos son suficientes para comenzar el desarrollo, se comenzará con la estructuración y organización de éstos.

La formalización de las necesidades se debe estructurar como sigue:

- Equipos reales que deben estar presentes en el banco.
- Referencias de documentación e información de equipos (tamaño, conectores...).
- Referencias de documentación de diagramas eléctricos.
- Referencias de ICD's (Mensajes, escalados, rangos...) y de elementos físicos (conectores, *pinout*...).
- Señales que deben ser estimuladas y/o monitorizadas.
- Capacidades de almacenamiento y grabación (*recording*) de señales (número de señales, velocidad, *throughput*...)
- Simulaciones necesarias de entorno, de avión o tácticas.
- Funciones no desarrolladas ya por el entorno del grupo de trabajo.
- Escenarios de uso del banco (combinaciones de equipos reales y simulados, conexiones con otros recursos, instrumentación especial...)
- Capacidades de conexión con otros bancos o recursos.
- Escenarios de test a realizar automáticamente.

La salida de esta actividad es el documento *Bench Requirements*. En este apartado se deben incluir *block diagrams* del banco, escenarios operativos, capacidad de conexiones y toda la información relevante.

### AVI.iv. Diseño preliminar

Tras evaluar las necesidades exigidas por el cliente, se debe ofrecer una solución técnica a las exigencias, describiendo detalladamente los bloques principales de HW y SW, sus conexiones y el *layout* del diseño.

La salida de este proceso será el documento *Bench Preliminary Design*. Una vez realizado el documento, debe ser verificado por el cliente.

El evento responsable de la realización del proyecto será la existencia del conjunto de necesidades validadas del cliente.

El final del proceso es la verificación del diseño preliminar por parte del cliente, y la solución de todas las discrepancias.

Actividades:

### III. Concretar el diseño preliminar de HW

Es necesario definir el número y tipo de *racks*, la colocación de los *racks*, exigencias de potencia y refrigeración de cada *rack*, configuración de los *patch panels* número y tipo de señales, prestando especial atención a las Analógicas-Discretas.

En este apartado se incluye un diagrama de alto nivel.

#### I. Definir la configuración del subsistema de ensayo

Se debe organizar la definición en un modelo estándar para que el grupo de desarrollo del subsistema pueda darn el soporte adecuado.

- Número de tarjetas, tipo de chasis<sup>5</sup> y el número de CPU's y paneles de entrada y salida (I/O) por chasis.
- Ordenadores de sobremesa y sistema operativo usado<sup>6</sup>.

---

<sup>5</sup> Los estándares soportados son PXI y VME.

- Herramientas de test necesarias (*Datavisualization, Sequence Definition, Recording Definition...*).

Se obtiene un documento de definición que será agregado al documento *Bench Preliminary Design*.

## II. Identificar señales

Localizar todas las señales que se deben incluir en los bancos. Se dividen en tres grupos:

- Señales de avión directamente importadas del ICD (*Bench A/C*). Se usarán buses de señal como AFDX, 1553, Arinc 429 and CAN.
- Señales *Bench Derived A/C*. Aquí están señales analógicas, discretas, RVDT, LVDT y de sincronismo. Los escalados dependen de la interfaz HW del banco.
- Señales del RIG empleadas para realizar operaciones del usuario.

De nuevo se obtiene un documento de especificaciones que se debe incluir en el *Bench Specification Document*.

## III. Identificar simulaciones

Para cada simulación, escenario, o equipo especificado hay que concretar las necesidades propias de ella y las particularidades informales.

Se adjunta el documento en el *Bench Specification Document*.

## IV. Identificar funciones especiales

Las capacidades no realizadas ya por el grupo de subsistema, que exijan manejo del panel de I/O deben ser descritas como necesidades SW.

---

<sup>6</sup> Capacidad de desarrollo en Windows y Linux.

V. Identificar escenarios de pruebas

Se definirán los escenarios de pruebas necesarios para que sean contemplados en el diseño preliminar del banco.

VI. Identificar instrumentación especial

Se estudiarán las necesidades instrumentales para poder contemplar la inclusión de instrumentación específica en el diseño preliminar.

**AVI.v. Implementación del Banco**

Una vez que el diseño preliminar está verificado, se desarrolla la implementación del banco. Esta implementación se llevará a cabo en 2 caminos, HW y SW:

Implementación HW

- Diseño detallado de cada *rack* y de la colocación de equipos.
- Construcción y prueba de cada *rack* y de su estructura.

Implementación SW

- Implementación de la capa SW.
- Configuración, colocación, publicación<sup>7</sup> y prueba del subsistema de ensayo.

Estas dos partes del proceso convergen en la implementación del banco completo.

---

<sup>7</sup> Se debe hacer visible el subsistema de ensayo desde todos los puntos que sea necesario. (Database Population)



El evento de disparo de este proceso es la existencia de un diseño preliminar verificado.

El fin del proceso se consigue con la verificación de la implementación HW/SW y la existencia del banco dispuesto para pasar el proceso de aceptación.

Actividades:

I. Realización de diseño detallado HW.

El diseño preliminar del banco se concreta ahora en documentos de fabricación y consecución:

- Dibujos de paneles, *patch-panels*, serigrafía y *trays*.
- Dibujos de cableados.
- Dibujos de interconexión de cabinas.
- Tarjetas de adaptación.

Estas tareas suelen estar subcontratadas, por lo que debemos verificar los documentos obtenidos entre el jefe de la subcontrata y el jefe de proyecto.

La documentación se incluirá en el *Bench Sepcification Document*.

II. Fabricar y obtener parte HW.

Esta tarea se realiza por la subcontrata, pero se supervisa por el jefe de proyecto.

Se debe realizar conforme al documento *Integration Benh HW guidelines*. Una vez realizado deberá ser verificado y validado tanto por el grupo de desarrollo del banco como por los fabricantes (subcontratados).

La documentación se incluirá en el *Bench Sepcification Document*.

### III. Realizar diseño del SW específico.

Cada banco tiene dos partes SW, la parte de subsistema de ensayo, y el SW específico del banco.

El SW de subsistema es responsabilidad del *Test System Group*, pero el SW específico del banco, como simulaciones, procesamiento de señales especiales o post-procesamiento de señales es responsabilidad del grupo de desarrollo de banco.

El diseño del SW específico debe respetar las normas del documento *Integration Bench SW guidelines* y las siguientes reglas:

- Cada componente SW desarrollado para un banco debe tener una especificación de necesidades, una especificación de diseño y una especificación de test.
- El código fuente y la documentación debe estar bajo control del grupo de desarrollo de banco (BDG) y archivado correctamente.
- Tras la entrega del banco al cliente, el código SW y los resultados de test deben ser verificados por el BDG.

### IV. Configurar subsistema de ensayo.

La configuración consta de los siguientes apartados:

- Instalación de procesadores y tarjetas de entrada salida.
- Configuración de las tarjetas.
- Configuración del software genérico de subsistema.
- Alta de usuarios.
- Instalación de las funciones especiales de SW.

La configuración debe ser verificada por pruebas, estimulación de todas las I/O y definición de señales.

La configuración debe especificar tanto la configuración HW como SW (posiciones de tarjetas, *drivers* usados...).

Estas tareas serán llevadas a cabo por el *Test System Department*.

V. Realizar integración del banco.

Antes de realizar la integración final del banco en el laboratorio, habrá que tener en cuenta los siguientes aspectos:

- Requerimientos de entorno como espacio para el banco, refrigeración... (*Systems Laboratories Dep.*)
- Capacidad LAN y WAN para el uso de VLAN por parte de los sistemas de pruebas (*Test System Dep., Infrastructure & Configuration Dep. and Product Group Leader*)
- Distribución suficiente de Potencia. Seguridad de la red, de la sala... (*Test System Dep., Systems Laboratories, Infrastructure & Configuration Dep., Product Group Leader and MTAD security Dept.*)

En el momento que el subsistema de ensayo, los *racks* y tarjetas y el SW específico están correctamente instalados, se comenzarán las pruebas, hasta que se considere que está completamente preparado para pasar el proceso de verificación.

Estas actividades serán supervisadas por el fabricante del banco. La aceptación del banco, forma parte del proceso de validación.

AVI.vi. Consecución y Subcontratación

Este proceso se realiza muchas veces durante el proceso de construcción del banco. La subcontratación se realiza en:

- Diseño detallado y fabricación de *racks* y *harnessing*.
- Material para los *racks* y *harnessing*.
- Publicación interna de las bases de datos de señales.
- Diseño y fabricación de la electrónica de adaptación de bancos.
- Simulaciones SW.

La condición de disparo del proceso es la aprobación de ejecución de la propuesta de banco, y para cada apartado se realiza cuando hay una lista de material necesario o una necesidad específica de subcontratación.

La salida de este proceso son los componentes HW/SW encargados o subcontratados.

El grupo subcontratado para la realización de las tareas debe seguir las actividades descritas en el proceso de implementación de banco en 1.3.4.

Actividades:

I. Subcontratación del diseño detallado de HW y su fabricación

El diseño detallado de los *racks*, ensamblado y fabricación de los mismos, debe ser subcontratado a un especialista en la construcción de los mismos.

El BDG y el grupo subcontratado de HW deben repasar conjuntamente:

- Diseño inicial del banco.
- Diseño detallado del HW del banco, incluyendo todas las partes involucradas.
- Resultados de las pruebas eléctricas.

Los documentos obtenidos en esta actividad deberán ser debidamente archivados en el *Test Development Site* (OWL)

## II. Subcontratación de simulaciones SW

Al igual que en la parte HW, las simulaciones SW deben ser encargados a especialistas cualificados para realizar simulaciones de sistemas integrados.

El BDG y el grupo subcontratado SW deben repasar conjuntamente:

- Requerimientos SW.
- Resultados de test SW.

De nuevo se almacenará debidamente la información en OWL.

## III. Consecución de los materiales del subsistema

Cuando el subsistema de ensayo está debidamente configurado, se realizará una lista con los materiales necesarios, que será pasado a la empresa encargada del suministro de material. Esta empresa informará sobre las fechas de las entregas.

El material recibido será debidamente revisado por el BDG, e instalado adecuadamente antes de ser enviado al grupo de fabricación del banco.

### **AVI.vii. Verificación**

La verificación se realiza también en los resultados intermedios (documentación y partes HW y SW) del desarrollo para comprobar la adecuación a las necesidades definidas por el cliente.

La verificación requiere 3 tipos de acciones:

- Inspección.

- Revisión.
- Pruebas.

Las discrepancias y problemas que surjan durante la verificación deben ser resueltas por el jefe del BDG.

El evento de disparo del proceso será la existencia de elementos que necesiten ser verificados.

El final del proceso será la producción de un documento de verificación.

Actividades:

#### I. Inspecciones

La documentación y las soluciones SW generadas durante las revisiones de necesidades, los diseños de alto nivel, los diseños detallados, la integración y la aceptación debe ser revisada por los autores de acuerdo a una *checklist*.

Los dibujos y planos de fabricación de *racks* y ensamblado deben ser vigilados por el diseñador y el BDG, para comprobar que sigue las líneas del diseño del banco.

Los productos como los *racks* y los ensamblados deben ser vigilado por el fabricante y el BDG para verificar la realización conforme al diseño de banco establecido.

#### II. Organización de reuniones de revisión

La documentación de necesidades, los diseños preliminares y los procedimientos de aceptación debe ser periódicamente revisado en reuniones por el cliente y el BDG. Estas revisiones estarán dirigidas por el jefe del BDG, y firmadas tanto por éste como por el jefe de la organización cliente.

Las reuniones deberán ser debidamente documentadas, mostrando las conclusiones, acciones y decisiones sobre las tareas.

### III. Realizar pruebas

Habrá que realizar diferentes pruebas sobre las partes del banco.

- Pruebas de SW:
  - Prueba de cada parte específica de SW, comprobando si cumple los requerimientos.
- Pruebas de HW:
  - Pruebas eléctricas del banco fabricado por la subcontrata.
- Pruebas de integración HW / SW
  - Subsistema de ensayo / Integración del banco

Para cada tipo de prueba se deberá obtener documentación de los casos de test y los resultados de éstos (*test cases / test results*) y almacenada debidamente en OWL.

### AVI.viii. Validación

La validación se realiza sobre las necesidades exigidas por el cliente correctamente formalizadas (la salida del proceso de análisis de necesidades) y sobre la implementación de éstas (después de la integración del proceso de implementación).

Esta validación se realizará entre el cliente y el BDG.

La condición de disparo del proceso será:

- Necesidad de validación de las necesidades disponibles con respecto a las necesidades exigidas.
- Necesidad de validación de la implementación final del banco.

La salida de este proceso será:

- Documento de validación de las necesidades.
- Documento de validación de la implementación del banco.

Actividades:

#### I. Validación de requerimientos

La validación de requerimientos se realizará después de la realización del proceso de análisis de necesidades, y posteriormente a las revisiones de estas necesidades entre clientes y BDG.

Para validación se emplea el mismo mecanismo que para verificación.

La revisión conjunta de necesidades se terminará antes de la finalización del diseño de alto nivel del banco. Este diseño puede empezar antes de finalizar el análisis de necesidades, pero no puede finalizar antes de acabar lo anterior.

La reunión de revisión de necesidades deberá estar documentada mostrando conclusiones, acciones y decisiones sobre las tareas.

El documento deberá ser firmado por los responsables del BDG y de la organización cliente, y será debidamente almacenado en OWL.

Cuando se termina con la validación de necesidades se realizará el diseño de tareas y la evaluación de riesgos.

#### II. Validación de la implementación



Cada banco debe ser validado con respecto a las necesidades del cliente, por lo que el proceso de validación deberá ser acordado por el BDG y el cliente.

El proceso de validación podrá incluir inspecciones, revisiones y pruebas. El documento obtenido en el proceso de validación deberá identificar los productos que están siendo validados:

- Configuración de subsistema de ensayo
- Especificaciones de banco
- Versiones del SW específico

Las pruebas deben estar basadas en las pruebas realizadas durante la integración del banco, pero ahora estarán supervisadas también por el cliente.

Cuando los requerimientos se van indicando progresivamente, la validación del banco debe estar relacionada con esta progresión, realizando documentos que indiquen el punto de requerimientos en que se enmarca.

#### **AVI.ix. Proceso de configuración**

El proceso de configuración incluye tres actividades independientes:

##### **I. Identificar la documentación realizada**

Todos los documentos realizados deben ser correctamente numerados según el “Sistema de numeración de la documentación de la DPS”.

Los planos eléctricos y mecánicos deben ser correctamente identificados de acuerdo con la “Numeración y Verificación de planos de las instalaciones de ensayo de IDS”.

Los elementos HW/SW producidos, el código fuente generado, ficheros “.uml”... debe ser correctamente identificado con respecto a la versión del banco.

## II. Controlar los cambios sobre el material producido

### a. Documentos

Cuando un documento es generado, se formalizará su identificación y cada cambio realizado será precisamente controlado. Para ello se utilizará el sistema CVS / SUBVERSION.

### b. SW y base de datos de señales

De nuevo emplearemos CVS / SUBVERSION para almacenar la documentación relacionada, indicando en cada caso el historial de cambios sobre el documento.

Generalmente los cambios en la base de datos de señales se producen por cambios en los ICD's, por lo que será necesario precisar a que ICD corresponde cada versión.

### c. Planos

Los planos deberán ser accesibles desde CAD tool, siendo el historial de navegación almacenado en Excel Book.

## III. Archivar y recuperar la documentación de configuración

Tras la validación del banco, toda la documentación generada deberá ser correctamente almacenada en un registro de documentación del banco, almacenado con la seguridad oportuna y copiado en soporte óptico que estará registrado.

- Especificación del banco
- Planos eléctricos y mecánicos
- Casos de prueba y resultados del proceso de validación
- Código fuente de las simulaciones
- Documentación de las simulaciones
- Base de datos de señales (ficheros “.xml”)

#### IV. Back-Ups

Se realizarán *back-up* incrementales periódicos de los discos del banco, exceptuando los clientes, y adicionalmente se realizará un *back-up* del banco completo mensualmente en un servidor exterior.

#### AVI.x. Documentación

Toda la documentación deberá ser identificado con el correspondiente NT y procesados de acuerdo con el procedimiento genérico de documentación de ingeniería del MTA, “Sistema de numeración de la documentación de la DPS”.

Todas las versiones de la documentación deberán ser actualizadas en CVS.

Para el correcto almacenamiento del documento se deberán seguir los siguientes criterios:

- Todos los documentos deben ser revisados y aprobados por el responsable del BDG.
- Los planes de calidad adicionales y las especificaciones de banco deberán ser aprobadas por el responsable del *Test System Dep.*
- La documentación del banco deberá ser guardada en OWL.

#### AVI.xi. Soporte

Después de la validación y entrega del banco, el responsable del BDG comenzará con el proceso de soporte para resolución de problemas. Este proceso manejará los *problem report* y cambios pedidos por el cliente.

Se dispondrá de una base de datos de problemas del banco manejada con PRMS para la correcta realización y seguimiento de los problemas o cambios pedidos por el cliente.

Los pasos a seguir son:

- Problema/Cambio asociado al subsistema de ensayo SW  
=>  
El *Test System Dep.* realizará las mejoras precisas en el subsistema, e instalará la nueva versión después de la validación.
- Problema/Cambio asociado al SW específico del Banco  
=>  
El grupo encargado producirá una nueva versión validada.
- Problema/Cambio asociado al HW del banco  
=>  
El responsable del BDG decidirá quién se hace cargo.

Cada modificación del banco HW/SW tendrá control sobre los cambios de configuración y actualizaciones sobre la documentación del banco.

Actividades:

I. Manejar problemas y actuaciones

Se tendrá una base de datos de los problemas de cada banco, gestionada con PRMS, que nos permitirá manejar y seguir cada problema o cambio recibido.

Cuando se acepta la resolución del problema se irá progresando a través de varios estados hasta que llega al estado *closed*.

## II. Dirigir el soporte

El responsable del BDG se encargará de empaquetar los problemas o cambios en grupos, de modo que la solución, las modificaciones de HW/SW y la verificación y validación sea más sencillo.

Durante esta fase los procesos SW y HW son de nuevo invocados y la configuración del banco modificada, además de actualizada la documentación referente al banco.

Cada vez que la configuración se modifica y es validada y entregada al cliente, se realizará nueva documentación que será entregada al *Configuration Dpt.*

## **Anexo VII. Resultado de pruebas del banco**

### **AVII.i. Introducción**

Se presentan a continuación los resultados obtenidos en la realización de las diferentes pruebas de verificación sobre el banco de ensayo.

### **AVII.ii. Resultados pruebas de continuidad**

- Mazo A4 M-A97

<b><u>MAZO M-A97</u></b>			<b><u>P/N: A4 M-A97</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
1 (IP14-J1)	✓		✓		✓	
2 (IP14-J1)	✓		✓		✓	
3 (IP14-J1)	✓		✓		✓	
4 (IP14-J1)	✓		✓		✓	
5 (IP14-J1)	✓		N/A	N/A	N/A	N/A
138 (TS31-J1)	N/A	N/A	✓		✓	
149 (TS31-J1)	N/A	N/A	✓		✓	
148 (TS31-J1)	N/A	N/A	✓		✓	

<b><u>MAZO M-A97</u></b>				<b><u>P/N: A4 M-A97</u></b>		
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
150 (TS31-J1)	N/A	N/A	N/A	N/A	✓	

Tabla 9 Resultados M-A97

- Mazo A4 M-A98

<b><u>MAZO M-A98</u></b>				<b><u>P/N: A4 M-A98</u></b>		
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
1 (IP14-J2)	✓		✓		✓	
2 (IP14-J2)	✓		✓		✓	
3 (IP14-J2)	✓		✓		✓	
4 (IP14-J2)	✓		✓		✓	
5 (IP14-J2)	✓		N/A	N/A	N/A	N/A
138 (TS32-J1)	N/A	N/A	✓		✓	
149 (TS32-J1)	N/A	N/A	✓		✓	
148 (TS32-J1)	N/A	N/A	✓		✓	

<b><u>MAZO M-A98</u></b>			<b><u>P/N: A4 M-A98</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
150 (TS32-J1)	N/A	N/A	N/A	N/A	✓	

Tabla 10 Resultados M-A98

- Mazo A4 M-A99

<b><u>MAZO M-A99</u></b>			<b><u>P/N: A4 M-A99</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
1 (IP14-J3)	✓		✓		✓	
2 (IP14-J3)	✓		✓		✓	
3 (IP14-J3)	✓		✓		✓	
4 (IP14-J3)	✓		✓		✓	
5 (IP14-J3)	✓		N/A	N/A	N/A	N/A
138 (TS33-J1)	N/A	N/A	✓		✓	
149 (TS33-J1)	N/A	N/A	✓		✓	
148 (TS33-J1)	N/A	N/A	✓		✓	



<b>MAZO M-A99</b>			<b>P/N: A4 M-A99</b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
150 (TS33-J1)	N/A	N/A	N/A	N/A	✓	

Tabla 11 Resultados M-A99

- Mazo A4 M-A100

<b>MAZO M-A100</b>			<b>P/N: A4 M-A100</b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
1 (IP14-J4)	✓		✓		✓	
2 (IP14-J4)	✓		✓		✓	
3 (IP14-J4)	✓		✓		✓	
4 (IP14-J4)	✓		✓		✓	
5 (IP14-J4)	✓		N/A	N/A	N/A	N/A
138 (TS34-J1)	N/A	N/A	✓		✓	
149 (TS34-J1)	N/A	N/A	✓		✓	
148 (TS34-J1)	N/A	N/A	✓		✓	

<b><u>MAZO M-A100</u></b>			<b><u>P/N: A4 M-A100</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
150 (TS34-J1)	N/A	N/A	N/A	N/A	✓	

Tabla 12 Resultados M-A100

- Mazo A4 M-A101

<b><u>MAZO M-A101</u></b>			<b><u>P/N: A4 M-A101</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
31 (SG20-J4)	✓		✓		✓	
30 (SG20-J4)	✓		✓		✓	
29 (SG20-J4)	✓		✓		✓	
28 (SG20-J4)	✓		✓		✓	
32 (SG20-J4)	✓		N/A	N/A	N/A	N/A
1 (IP14-J6)	N/A	N/A	✓		✓	
2 (IP14-J6)	N/A	N/A	✓		✓	
3 (IP14-J6)	✓		✓		✓	

<b>MAZO M-A101</b>			<b>P/N: A4 M-A101</b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
4 (IP14-J6)	✓		✓		✓	
5 (IP14-J6)	N/A	N/A	✓		N/A	N/A
6 (TS22-X5)	N/A	N/A	N/A	N/A	✓	
42 (TS22-X5)	N/A	N/A	N/A	N/A	✓	
1 (IP14-J7)	N/A	N/A	✓		✓	
2 (IP14-J7)	N/A	N/A	✓		✓	
3 (IP14-J7)	✓		✓		✓	
4 (IP14-J7)	✓		✓		✓	
5 (IP14-J7)	N/A	N/A	✓		N/A	N/A
6 (TS23-X5)	N/A	N/A	N/A	N/A	✓	
42 (TS23-X5)	N/A	N/A	N/A	N/A	✓	

Tabla 13 Resultados M-A101

- Mazo A4 M-A08

<b><u>MAZO M-A08</u></b>			<b><u>P/N: A4 M-A08</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
A (PW17-J6)	✓		✓		✓	
B (PW17-J6)	✓		✓		✓	
C (PW17-J6)	✓		✓		✓	
D (PW17-J6)	✓		✓		✓	
E (PW17-J6)	✓		✓		✓	
F (PW17-J6)	✓		✓		✓	
G (PW17-J6)	✓		✓		✓	
H (PW17-J6)	✓		✓		✓	
A (CP11-X5)	N/A	N/A	✓		✓	
B (CP11-X5)	N/A	N/A	✓		✓	
F (CP11-X5)	✓		N/A	N/A	N/A	N/A
1 (CP12-X1)	N/A	N/A	✓		✓	

<b><u>MAZO M-A08</u></b>			<b><u>P/N: A4 M-A08</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
2 (CP12-X1)	N/A	N/A	✓		✓	
5 (CP12-X1)	N/A	N/A	✓		✓	
9 (CP12-X1)	✓		N/A	N/A	N/A	N/A
A (IP05-J2)	N/A	N/A	✓		✓	
B (IP05-J2)	N/A	N/A	✓		✓	
C (IP05-J2)	N/A	N/A	✓		✓	
D (IP05-J2)	✓		N/A	N/A	N/A	N/A

Tabla 14 Resultados M-A08

- Mazo A4 M-A10

<b><u>MAZO M-A10</u></b>			<b><u>P/N: A4 M-A10</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
35 (IP05-J4)	✓		✓		✓	
36 (IP05-J4)	✓		✓		✓	

<b><u>MAZO M-A10</u></b>			<b><u>P/N: A4 M-A10</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
37 (IP05-J4)	✓		✓		✓	
38 (IP05-J4)	✓		✓		✓	
39 (IP05-J4)	✓		✓		✓	
40 (IP05-J4)	✓		N/A	N/A	N/A	N/A
41 (IP05-J4)	✓		✓		✓	
46 (IP05-J4)	✓		✓		✓	
42 (IP05-J4)	✓		✓		✓	
45 (IP05-J4)	✓		✓		✓	
47 (IP05-J4)	✓		✓		✓	
48 (IP05-J4)	✓		✓		✓	
43 (IP05-J4)	✓		✓		✓	
44 (IP05-J4)	✓		✓		✓	

<b>MAZO M-A10</b>			<b>P/N: A4 M-A10</b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b> AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
123 (IP05-J3)	N/A	N/A	✓		✓	
3 (CP12-X3)	N/A	N/A	✓		✓	
1 (CP12-X3)	N/A	N/A	✓		✓	
C (CP11-X9)	N/A	N/A	✓		✓	
A (CP11-X9)	N/A	N/A	✓		✓	
4 (CP12-X1)	N/A	N/A	✓		✓	
6 (CP12-X1)	N/A	N/A	✓		✓	
7 (CP12-X1)	N/A	N/A	✓		✓	
8 (CP12-X1)	N/A	N/A	✓		✓	
34 (CP12-X3)	N/A	N/A	✓		✓	
35 (CP12-X3)	N/A	N/A	✓		✓	
D (CP11-X5)	N/A	N/A	✓		✓	

<b><u>MAZO M-A10</u></b>			<b><u>P/N: A4 M-A10</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
E (CP11-X5)	N/A	N/A	N/A	N/A	✓	

Tabla 15 Resultados M-A10

- Mazo A4 M-A14

<b><u>MAZO M-A14</u></b>			<b><u>P/N: A4 M-A14</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
A (CP12-X2)	✓		✓		✓	
H (CP12-X2)	✓		✓		✓	
F (CP12-X2)	✓		✓		✓	
S (CP12-X2)	✓		✓		✓	
L (CP12-X2)	✓		✓		✓	
T (CP12-X2)	✓		✓		✓	
M (CP12-X2)	✓		✓		✓	
C-HI (CP12-X2)	✓		✓		✓	



<b>MAZO M-A14</b>			<b>P/N: A4 M-A14</b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b> AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
C-LO (CP12-X2)	✓		✓		✓	
C-HI (CP12-X2)	✓		✓		✓	
C-LO (CP12-X2)	✓		✓		✓	
49 (IP05-J4)	N/A	N/A	✓		✓	
50 (IP05-J4)	N/A	N/A	✓		✓	
51 (IP05-J4)	N/A	N/A	✓		✓	
52 (IP05-J4)	N/A	N/A	✓		✓	
53 (IP05-J4)	N/A	N/A	✓		✓	
54 (IP05-J4)	N/A	N/A	✓		✓	
55 (IP05-J4)	N/A	N/A	✓		✓	
C (ZC09-P1)	N/A	N/A	✓		✓	
I (ZC09-P1)	N/A	N/A	✓		✓	

<b><u>MAZO M-A14</u></b>			<b><u>P/N: A4 M-A14</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
C (ZC09-P2)	N/A	N/A	✓		✓	
I (ZC09-P2)	N/A	N/A	N/A	N/A	✓	

Tabla 16 Resultados M-A14

- Mazo A4 M-A15

<b><u>MAZO M-A15</u></b>			<b><u>P/N: A4 M-A15</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
2 (CP12-X3)	✓		✓		✓	
4 (CP12-X3)	✓		✓		✓	
5 (CP12-X3)	✓		✓		✓	
6 (CP12-X3)	✓		✓		✓	
7 (CP12-X3)	✓		✓		✓	
8 (CP12-X3)	✓		✓		✓	
9 (CP12-X3)	✓		✓		✓	

<b><u>MAZO M-A15</u></b>			<b><u>P/N: A4 M-A15</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
10 (CP12-X3)	✓		✓		✓	
11 (CP12-X3)	✓		✓		✓	
12 (CP12-X3)	✓		✓		✓	
13 (CP12-X3)	✓		✓		✓	
14 (CP12-X3)	✓		✓		✓	
15 (CP12-X3)	✓		✓		✓	
16 (CP12-X3)	✓		✓		✓	
17 (CP12-X3)	✓		✓		✓	
18 (CP12-X3)	✓		✓		✓	
19 (CP12-X3)	✓		✓		✓	
20 (CP12-X3)	✓		✓		✓	
21 (CP12-X3)	✓		✓		✓	

<b><u>MAZO M-A15</u></b>			<b><u>P/N: A4 M-A15</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
22 (CP12-X3)	✓		✓		✓	
23 (CP12-X3)	✓		✓		✓	
24 (CP12-X3)	✓		✓		✓	
29 (CP12-X3)	✓		✓		✓	
30 (CP12-X3)	✓		✓		✓	
31 (CP12-X3)	✓		✓		✓	
32 (CP12-X3)	✓		✓		✓	
33 (CP12-X3)	✓		✓		✓	
B (CP11-X9)	N/A	N/A	✓		✓	
D (CP11-X9)	N/A	N/A	✓		✓	
E (CP11-X9)	N/A	N/A	✓		✓	
F (CP11-X9)	N/A	N/A	✓		✓	

<b><u>MAZO M-A15</u></b>			<b><u>P/N: A4 M-A15</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
G (CP11-X9)	N/A	N/A	✓		✓	
H (CP11-X9)	N/A	N/A	✓		✓	
J (CP11-X9)	N/A	N/A	✓		✓	
K (CP11-X9)	N/A	N/A	✓		✓	
L (CP11-X9)	N/A	N/A	✓		✓	
M (CP11-X9)	N/A	N/A	✓		✓	
N (CP11-X9)	N/A	N/A	✓		✓	
P (CP11-X9)	N/A	N/A	✓		✓	
R (CP11-X9)	N/A	N/A	✓		✓	
S (CP11-X9)	N/A	N/A	✓		✓	
T (CP11-X9)	N/A	N/A	✓		✓	
U (CP11-X9)	N/A	N/A	✓		✓	

<b><u>MAZO M-A15</u></b>			<b><u>P/N: A4 M-A15</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
V (CP11-X9)	N/A	N/A	✓		✓	
W (CP11-X9)	N/A	N/A	✓		✓	
X (CP11-X9)	N/A	N/A	✓		✓	
Y (CP11-X9)	N/A	N/A	✓		✓	
Z (CP11-X9)	N/A	N/A	✓		✓	
A (CP11-X9)	N/A	N/A	✓		✓	
f (CP11-X9)	N/A	N/A	✓		✓	
g (CP11-X9)	N/A	N/A	✓		✓	
h (CP11-X9)	N/A	N/A	✓		✓	
i (CP11-X9)	N/A	N/A	✓		✓	
j (CP11-X9)	N/A	N/A	N/A	N/A	✓	

Tabla 17 Resultados M-A15

- Mazo A4 M-A16

<b>MAZO M-A16</b>			<b>P/N: A4 M-A16</b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
A (CP11-X2)	✓		✓		✓	
B (CP11-X2)	✓		✓		✓	
C (CP11-X2)	✓		✓		✓	
D (CP11-X2)	✓		✓		✓	
E (CP11-X2)	✓		✓		✓	
5 (DB9 AERIAL CONN.)	N/A	N/A	✓		✓	
7 (DB9 AERIAL CONN.)	N/A	N/A	✓		✓	
3 (DB9 AERIAL CONN.)	N/A	N/A	✓		✓	
2 (DB9 AERIAL CONN.)	N/A	N/A	✓		✓	
8 (DB9 AERIAL CONN.)	N/A	N/A	✓		✓	
1 (DB9 AERIAL CONN.)	✓		N/A	N/A	N/A	N/A

Tabla 18 Resultados M-A16

- Mazo A4 M-A17

<b><u>MAZO M-A17</u></b>			<b><u>P/N: A4 M-A17</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
G (CP11-X8)	✓		✓		✓	
P (CP11-X8)	✓		✓		✓	
H (CP11-X8)	✓		✓		✓	
R (CP11-X8)	✓		✓		✓	
J (CP11-X8)	✓		✓		✓	
S (CP11-X8)	✓		✓		✓	
K (CP11-X8)	✓		✓		✓	
T (CP11-X8)	✓		✓		✓	
L (CP11-X8)	✓		✓		✓	
U (CP11-X8)	✓		✓		✓	
M (CP11-X8)	✓		✓		✓	
V (CP11-X8)	✓		✓		✓	



<b><u>MAZO M-A17</u></b>			<b><u>P/N: A4 M-A17</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
N (CP11-X8)	✓		✓		✓	
A (CP11-X8)	✓		✓		✓	
B (CP11-X8)	✓		✓		✓	
E (CP11-X8)	✓		✓		✓	
1 (IP14-J1)	N/A	N/A	✓		✓	
2 (IP14-J1)	N/A	N/A	✓		✓	
3 (IP14-J1)	N/A	N/A	✓		✓	
4 (IP14-J1)	N/A	N/A	✓		✓	
5 (IP14-J1)	✓		N/A	N/A	N/A	N/A
1 (IP14-J2)	N/A	N/A	✓		✓	
2 (IP14-J2)	N/A	N/A	✓		✓	
3 (IP14-J2)	N/A	N/A	✓		✓	

<b><u>MAZO M-A17</u></b>			<b><u>P/N: A4 M-A17</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
4 (IP14-J2)	N/A	N/A	✓		✓	
5 (IP14-J2)	✓		N/A	N/A	N/A	N/A
1 (IP14-J3)	N/A	N/A	✓		✓	
2 (IP14-J3)	N/A	N/A	✓		✓	
3 (IP14-J3)	N/A	N/A	✓		✓	
4 (IP14-J3)	N/A	N/A	✓		✓	
5 (IP14-J3)	✓		N/A	N/A	N/A	N/A
1 (IP14-J4)	N/A	N/A	✓		✓	
2 (IP14-J4)	N/A	N/A	✓		✓	
3 (IP14-J4)	N/A	N/A	✓		✓	
4 (IP14-J4)	N/A	N/A	✓		✓	
5 (IP14-J4)	✓		N/A	N/A	N/A	N/A

<b><u>MAZO M-A17</u></b>			<b><u>P/N: A4 M-A17</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
1 (IP14-J6)	N/A	N/A	✓		✓	
2 (IP14-J6)	N/A	N/A	✓		✓	
3 (IP14-J6)	N/A	N/A	✓		✓	
4 (IP14-J6)	N/A	N/A	✓		✓	
5 (IP14-J6)	✓		N/A	N/A	N/A	N/A
1 (IP14-J7)	N/A	N/A	✓		✓	
2 (IP14-J7)	N/A	N/A	✓		✓	
3 (IP14-J7)	N/A	N/A	✓		✓	
4 (IP14-J7)	N/A	N/A	✓		✓	
5 (IP14-J7)	✓		N/A	N/A	N/A	N/A
1 (IP14-J5)	N/A	N/A	✓		✓	
2 (IP14-J5)	N/A	N/A	✓		✓	

<b><u>MAZO M-A17</u></b>			<b><u>P/N: A4 M-A17</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
3 (IP14-J5)	N/A	N/A	✓		✓	
4 (IP14-J5)	N/A	N/A	✓		✓	
5 (IP14-J5)	✓		N/A	N/A	N/A	N/A
1 (IP14-J8)	N/A	N/A	✓		✓	
2 (IP14-J8)	N/A	N/A	✓		✓	
3 (IP14-J8)	✓		N/A	N/A	N/A	N/A

Tabla 19 Resultados M-A17

- Mazo A4 M-A04

<b><u>MAZO M-A04</u></b>			<b><u>P/N: A4 M-A04</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
35 (IP06-J2)	✓		✓		✓	
36 (IP06-J2)	✓		✓		✓	
37 (IP06-J2)	✓		✓		✓	

<b><u>MAZO M-A04</u></b>			<b><u>P/N: A4 M-A04</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
38 (IP06-J2)	✓		✓		✓	
39 (IP06-J2)	✓		✓		✓	
40 (IP06-J2)	✓		✓		✓	
41 (IP06-J2)	✓		✓		✓	
42 (IP06-J2)	✓		✓		✓	
43 (IP06-J2)	✓		✓		✓	
44 (IP06-J2)	✓		✓		✓	
45 (IP06-J2)	✓		✓		✓	
46 (IP06-J2)	✓		✓		✓	
47 (IP06-J2)	✓		✓		✓	
48 (IP06-J2)	✓		✓		✓	
49 (IP06-J2)	✓		✓		✓	

<b><u>MAZO M-A04</u></b>			<b><u>P/N: A4 M-A04</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b> AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
50 (IP06-J2)	✓		✓		✓	
51 (IP06-J2)	✓		✓		✓	
52 (IP06-J2)	✓		✓		✓	
53 (IP06-J2)	✓		✓		✓	
54 (IP06-J2)	✓		✓		✓	
55 (IP06-J2)	✓		✓		✓	
17 (SG20-J2)	N/A	N/A	✓		✓	
33 (SG20-J4)	N/A	N/A	✓		✓	
34 (SG20-J4)	N/A	N/A	✓		✓	
35 (SG20-J4)	N/A	N/A	✓		✓	
36 (SG20-J4)	N/A	N/A	✓		✓	
37 (SG20-J4)	N/A	N/A	✓		✓	

<b><u>MAZO M-A04</u></b>			<b><u>P/N: A4 M-A04</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
38 (SG20-J4)	N/A	N/A	✓		✓	
39 (SG20-J4)	N/A	N/A	✓		✓	
40 (SG20-J4)	N/A	N/A	✓		✓	
41 (SG20-J4)	N/A	N/A	✓		✓	
42 (SG20-J4)	N/A	N/A	✓		✓	
43 (SG20-J4)	N/A	N/A	✓		✓	
44 (SG20-J4)	N/A	N/A	✓		✓	
45 (SG20-J4)	N/A	N/A	✓		✓	
46 (SG20-J4)	N/A	N/A	✓		✓	
47 (SG20-J4)	N/A	N/A	✓		✓	
48 (SG20-J4)	N/A	N/A	✓		✓	
49 (SG20-J4)	N/A	N/A	✓		✓	

<b><u>MAZO M-A04</u></b>			<b><u>P/N: A4 M-A04</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
50 (SG20-J4)	N/A	N/A	✓		✓	
51 (SG20-J4)	N/A	N/A	✓		✓	
52 (SG20-J4)	N/A	N/A	N/A	N/A	✓	

Tabla 20 Resultados M-A04

- Mazo A4 M-A07

<b><u>MAZO M-A07</u></b>			<b><u>P/N: A4 M-A07</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
10 (IP06-J6)	✓		✓		✓	
11 (IP06-J6)	✓		✓		✓	
12 (IP06-J6)	✓		✓		✓	
13 (IP06-J6)	✓		✓		✓	
14 (IP06-J6)	✓		N/A	N/A	N/A	N/A
19 (IP06-J6)	✓		✓		✓	



<b><u>MAZO M-A07</u></b>			<b><u>P/N: A4 M-A07</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
20 (IP06-J6)	✓		✓		✓	
21 (IP06-J6)	✓		✓		✓	
22 (IP06-J6)	✓		✓		✓	
23 (IP06-J6)	✓		N/A	N/A	N/A	N/A
24 (IP06-J6)	✓		✓		✓	
25 (IP06-J6)	✓		✓		✓	
26 (IP06-J6)	✓		✓		✓	
27 (IP06-J6)	✓		✓		✓	
28 (IP06-J6)	✓		✓		✓	
29 (IP06-J6)	✓		✓		✓	
30 (IP06-J6)	✓		✓		✓	
31 (IP06-J6)	✓		✓		✓	

<b><u>MAZO M-A07</u></b>			<b><u>P/N: A4 M-A07</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
32 (IP06-J6)	✓		N/A	N/A	N/A	N/A
1 (SG20-J4)	✓		✓		✓	
2 (SG20-J4)	✓		✓		✓	
3 (SG20-J4)	✓		✓		✓	
4 (SG20-J4)	✓		✓		✓	
5 (SG20-J4)	✓		✓		✓	
6 (SG20-J4)	✓		✓		✓	
7 (SG20-J4)	✓		✓		✓	
8 (SG20-J4)	✓		✓		✓	
9 (SG20-J4)	✓		✓		✓	
10 (SG20-J4)	✓		✓		✓	
11 (SG20-J4)	✓		✓		✓	

<b><u>MAZO M-A07</u></b>			<b><u>P/N: A4 M-A07</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
12 (SG20-J4)	✓		✓		✓	
13 (SG20-J4)	✓		✓		✓	
14 (SG20-J4)	✓		✓		✓	
15 (SG20-J4)	✓		✓		✓	
16 (SG20-J4)	✓		✓		✓	
17 (SG20-J4)	✓		✓		✓	
18 (SG20-J4)	✓		✓		✓	
19 (SG20-J4)	✓		✓		✓	
20 (SG20-J4)	✓		✓		✓	
24 (SG20-J4)	✓		✓		✓	
25 (SG20-J4)	✓		✓		✓	
26 (SG20-J4)	✓		✓		✓	

<b><u>MAZO M-A07</u></b>			<b><u>P/N: A4 M-A07</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
27 (SG20-J4)	✓		✓		✓	
32 (SG20-J4)	✓		N/A	N/A	N/A	N/A
25 (TS22-X3)	N/A	N/A	✓		✓	
17 (TS22-X3)	N/A	N/A	✓		✓	
33 (TS22-X3)	N/A	N/A	✓		✓	
47 (TS22-X3)	N/A	N/A	✓		✓	
25 (TS23-X3)	N/A	N/A	✓		✓	
17 (TS23-X3)	N/A	N/A	✓		✓	
33 (TS23-X3)	N/A	N/A	✓		✓	
47 (TS23-X3)	N/A	N/A	✓		✓	
24 (TS22-X5)	N/A	N/A	✓		✓	
31 (TS22-X5)	N/A	N/A	✓		✓	

<b><u>MAZO M-A07</u></b>			<b><u>P/N: A4 M-A07</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b> AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
9 (TS22-X5)	N/A	N/A	✓		✓	
16 (TS22-X5)	N/A	N/A	✓		✓	
55 (TS22-X5)	N/A	N/A	✓		✓	
54 (TS22-X5)	N/A	N/A	✓		✓	
53 (TS22-X5)	N/A	N/A	✓		✓	
48 (TS22-X5)	N/A	N/A	✓		✓	
45 (TS22-X5)	N/A	N/A	✓		✓	
37 (TS22-X5)	N/A	N/A	✓		✓	
4 (TS22-X5)	N/A	N/A	✓		✓	
10 (TS22-X5)	N/A	N/A	✓		✓	
12 (TS22-X5)	N/A	N/A	✓		✓	
34 (TS22-X5)	N/A	N/A	✓		✓	

<b><u>MAZO M-A07</u></b>			<b><u>P/N: A4 M-A07</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
24 (TS22-X5)	N/A	N/A	✓		✓	
31 (TS22-X5)	N/A	N/A	✓		✓	
9 (TS22-X5)	N/A	N/A	✓		✓	
16 (TS22-X5)	N/A	N/A	✓		✓	
55 (TS22-X5)	N/A	N/A	✓		✓	
54 (TS22-X5)	N/A	N/A	✓		✓	
53 (TS22-X5)	N/A	N/A	✓		✓	
48 (TS22-X5)	N/A	N/A	✓		✓	
45 (TS22-X5)	N/A	N/A	✓		✓	
37 (TS22-X5)	N/A	N/A	✓		✓	
4 (TS22-X5)	N/A	N/A	✓		✓	
10 (TS22-X5)	N/A	N/A	✓		✓	

<u>MAZO M-A07</u>			<u>P/N: A4 M-A07</u>			
	CONTINUIDAD		CONT. CRUZADA		AISLAMIENTO	
PIN (CONECTOR)	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
12 (TS22-X5)	N/A	N/A	✓		✓	
34 (TS22-X5)	N/A	N/A	N/A	N/A	✓	

Tabla 21 Resultados M-A07

- Mazo A4 M-A22

<u>MAZO M-A22</u>			<u>P/N: A4 M-A22</u>			
	CONTINUIDAD		CONT. CRUZADA		AISLAMIENTO	
PIN (CONECTOR)	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
A-c (TS22-X44)	✓		✓		✓	
A-i (TS22-X44)	✓		✓		✓	
B-c (TS22-X44)	✓		✓		✓	
B-i (TS22-X44)	✓		✓		✓	
C (TEE15-J3)	N/A	N/A	✓		✓	
I (TEE15-J3)	N/A	N/A	✓		✓	
C (TEE16-J3)	N/A	N/A	✓		✓	

<b><u>MAZO M-A22</u></b>			<b><u>P/N: A4 M-A22</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
I (TEE16-J3)	N/A	N/A	N/A	N/A	✓	

Tabla 22 Resultados M-A22

- Mazo A4 M-A23

<b><u>MAZO M-A23</u></b>			<b><u>P/N: A4 M-A23</u></b>			
	<b>CONTINUIDAD</b>		<b>CONT. CRUZADA</b>		<b>AISLAMIENTO</b>	
<b>PIN (CONECTOR)</b>	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
A-c (TS23-X44)	✓		✓		✓	
A-i (TS23-X44)	✓		✓		✓	
B-c (TS23-X44)	✓		✓		✓	
B-i (TS23-X44)	✓		✓		✓	
C (TEE17-J3)	N/A	N/A	✓		✓	
I (TEE17-J3)	N/A	N/A	✓		✓	
C (TEE18-J3)	N/A	N/A	✓		✓	
I (TEE18-J3)	N/A	N/A	N/A	N/A	✓	



Tabla 23 Resultados M-A23

- Mazo A4-2A122-00 M-A24

<u>MAZO M-A24</u>			<u>P/N: A4-2A122-00 M-A24</u>			
	CONTINUIDAD		CONT. CRUZADA		AISLAMIENTO	
PIN (CONECTOR)	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
A (IP06-J5)	✓		✓		✓	
B (IP06-J5)	✓		✓		✓	
C (IP06-J5)	✓		✓		✓	
D (IP06-J5)	✓		✓		✓	
1 (TS22-X13)	N/A	N/A	✓		✓	
4 (TS22-X13)	N/A	N/A	✓		✓	
1 (TS23-X13)	N/A	N/A	✓		✓	
4 (TS23-X13)	N/A	N/A	N/A	N/A	✓	

Tabla 24 Resultados M-A24

- Mazo A4 M-A96

<u>MAZO M-A96</u>	<u>P/N: A4 M-A96</u>
-------------------	----------------------

PIN (CONECTOR)	CONTINUIDAD		CONT. CRUZADA		AISLAMIENTO	
	PASA	NO PASA	PASA	NO PASA	PASA	NO PASA
1 (IP14-J5)	✓		✓		✓	
2 (IP14-J5)	✓		✓		✓	
3 (IP14-J5)	✓		✓		✓	
4 (IP14-J5)	✓		✓		✓	
5 (IP14-J5)	✓		N/A	N/A	N/A	N/A
5 (LVT-J4)	N/A	N/A	✓		✓	
1 (LVT -J4)	N/A	N/A	✓		✓	
4 (LVT -J4)	N/A	N/A	✓		✓	
3 (LVT -J4)	N/A	N/A	N/A	N/A	✓	

Tabla 25 Resultados M-A96

**AVII.iii. Resultados pruebas de funcionalidad**

- Prueba de inspección visual

<b><u>SUBRACK 1</u></b>		<b>P/N: A4 S/N: 01</b>
<b>PROCEDIMIENTO DE PRUEBA</b>	<b>PASA</b>	<b>NO PASA</b>
Comprobación del acabado superficial	✓	
Comprobación de la serigrafía	✓	
Comprobación del montaje mecánico	✓	
Comprobación de la correcta fijación de los componentes en el subrack	✓	
Comprobación del grapado/soldado de componentes	✓	
Comprobación de la correspondencia de galgas/cables	✓	
Comprobación de la correspondencia entre conectores y subrack	✓	
Comprobación del correcto etiquetado	✓	
Comprobación de dimensiones	✓	

**Tabla 26 Resultados visuales "Subrack 1"**

- Prueba funcional

SUBRACK 1 P/N: A4 S/N: 01			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
<b>PASO 1:</b> - Configurar el multímetro digital en función de medición de continuidad. - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J2/1 J4/1	No continuidad	✓
	J2/2 J4/2	No continuidad	✓
	J2/3 J4/3	No continuidad	✓
	J2/4 J4/4	No continuidad	✓
	J4/1 J3/9	Continuidad	✓
	J4/2 J3/10	Continuidad	✓
	J4/3 J3/11	Continuidad	✓
	J4/4 J3/12	Continuidad	✓
	J2/5 J4/5	No continuidad	✓
	J2/6 J4/6	No continuidad	✓
	J2/7 J4/7	No continuidad	✓
	J2/8 J4/8	No continuidad	✓
	J2/5 J3/15	No continuidad	✓
	J2/6 J3/16	No continuidad	✓

SUBRACK 1 P/N: A4 S/N: 01			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
	J2/7 J3/13	No continuidad	✓
	J2/8 J3/14	No continuidad	✓
	J4/5 J4/6	No continuidad	✓
	J4/7 J4/8	No continuidad	✓
<b>PASO 2:</b>  - Colocar los puentes de las minihembrillas PS04-PS03, PS36-PS35, PS12-PS11 y PS44-PS43.  - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J2/1 J4/1	Continuidad	✓
	J2/2 J4/2	Continuidad	✓
	J2/3 J4/3	Continuidad	✓
	J2/4 J4/4	Continuidad	✓
<b>PASO 3:</b>  - Colocar los puentes de las minihembrillas PS17-PS18, PS49-PS50, PS19-PS20 y PS51-PS52.  - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/5 J4/6	No continuidad	✓
	J4/7 J4/8	No continuidad	✓
<b>PASO 4:</b>	J4/5 J4/6	Continuidad	✓

<b>SUBRACK 1</b> <b>P/N: A4 S/N: 01</b>			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
<ul style="list-style-type: none"> <li>- Conmutar a la posición ON los interruptores SW01 y SW02.</li> <li>- Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.</li> </ul>	J4/7	Continuidad	✓
	J4/8		
<b>PASO 5:</b> <ul style="list-style-type: none"> <li>- Retirar los puentes de las minihembrillas PS17-PS18, PS49-PS50, PS19-PS20 y PS51-PS52.</li> <li>- Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.</li> </ul>	J4/5	No continuidad	✓
	J4/6		
<b>PASO 6:</b> <ul style="list-style-type: none"> <li>- Colocar los puentes de las minihembrillas PS17-PS18, PS49-PS50, PS19-PS20 y PS51-PS52.</li> <li>- Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.</li> </ul>	J4/7	No continuidad	✓
	J4/8		
	J4/5	No continuidad	✓
	J2/5		
	J4/6	No continuidad	✓
	J2/6		
	J4/7	No continuidad	✓
	J2/7		
	J4/8	No continuidad	✓
	J2/8		
	J4/5	No continuidad	✓
	J3/15		
	J4/6	No continuidad	✓
	J3/16		
	J4/7	No continuidad	✓
	J3/13		
	J4/8	No continuidad	✓
	J3/14		

SUBRACK 1 P/N: A4 S/N: 01			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
<b>PASO 7:</b> - Conmutar a la posición SEAS los interruptores SW06 y SW07. - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/5 J2/5	Continuidad	✓
	J4/6 J2/6	Continuidad	✓
	J4/7 J2/7	Continuidad	✓
	J4/8 J2/8	Continuidad	✓
	J4/5 J3/15	No continuidad	✓
	J4/6 J3/16	No continuidad	✓
	J4/7 J3/13	No continuidad	✓
	J4/8 J3/14	No continuidad	✓
<b>PASO 8:</b> - Conmutar a la posición EAMU los interruptores SW06 y SW07. - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/5 J2/5	No continuidad	✓
	J4/6 J2/6	No continuidad	✓
	J4/7 J2/7	No continuidad	✓
	J4/8 J2/8	No continuidad	✓
	J4/5 J3/15	Continuidad	✓
	J4/6 J3/16	Continuidad	✓

SUBRACK 1 P/N: A4 S/N: 01			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
	J4/7 J3/13	Continuidad	✓
	J4/8 J3/14	Continuidad	✓
<b>PASO 9:</b> - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/9 J3/1	Continuidad	✓
	J4/10 J3/2	Continuidad	✓
	J4/11 J3/3	Continuidad	✓
	J4/12 J3/4	Continuidad	✓
	J4/13 J3/5	Continuidad	✓
	J4/14 J3/6	Continuidad	✓
	J4/15 J3/7	Continuidad	✓
	J4/16 J3/8	Continuidad	✓
	J4/9 J2/9	No continuidad	✓
	J4/10 J2/10	No continuidad	✓
	J4/11 J2/11	No continuidad	✓
	J4/12 J2/12	No continuidad	✓



SUBRACK 1 P/N: A4 S/N: 01			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
	J4/13 J2/13	No continuidad	✓
	J4/14 J2/14	No continuidad	✓
	J4/15 J2/15	No continuidad	✓
	J4/16 J2/16	No continuidad	✓
	J4/17 J4/18	Continuidad	✓
	J4/19 J4/20	Continuidad	✓
	J4/17 J1/A	No continuidad	✓
	J4/18 J1/B	No continuidad	✓
	J4/19 J1/A	No continuidad	✓
	J4/20 J1/B	No continuidad	✓
<b>PASO 10:</b> - Colocar los puentes de las minihembrillas PS01-PS02, PS33-PS34, PS05-PS06, PS37-PS38, PS09-PS10, PS41-PS42, PS13-PS14 y PS45-PS46. - Conmutar a la posición CLEAR los	J4/9 J2/9	Continuidad	✓
	J4/10 J2/10	Continuidad	✓
	J4/11 J2/11	Continuidad	✓
	J4/12 J2/12	Continuidad	✓

SUBRACK 1 P/N: A4 S/N: 01			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
interruptores SW03 y SW04. - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/13 J2/13	Continuidad	✓
	J4/14 J2/14	Continuidad	✓
	J4/15 J2/15	Continuidad	✓
	J4/16 J2/16	Continuidad	✓
	J4/17 J4/18	No continuidad	✓
	J4/19 J4/20	No continuidad	✓
	J4/17 J1/A	Continuidad	✓
	J4/18 J1/B	Continuidad	✓
	J4/19 J1/A	Continuidad	✓
	J4/20 J1/B	Continuidad	✓
	J4/24 J4/31	Continuidad	✓
	J4/25 J4/30	No continuidad	✓
	J4/26 J4/29	Continuidad	✓
	J4/27 J4/28	No continuidad	✓

SUBRACK 1 P/N: A4 S/N: 01			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
	J4/33	No	✓
	J4/36	continuidad	
	J4/34	No	✓
	J4/35	continuidad	
	J4/38	Continuidad	✓
	J4/39		
<b>PASO 11:</b>  - Colocar los puentes de las minihembrillas PS39-PS40, PS47-PS48, PS21-PS22, PS53-PS54, PS23-PS24 y PS25-PS26.  - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/38	No	✓
	J2/17	continuidad	
	J4/38	No	✓
	J4/40	continuidad	
	J4/25	Continuidad	✓
	J4/30		
	J4/27	Continuidad	✓
	J4/28		
	J4/33	Continuidad	✓
	J4/36		
	J4/34	Continuidad	✓
	J4/35		
	J4/38	Continuidad	✓
	J2/17		
	J4/38	Continuidad	✓
	J4/40		
	J4/41	Continuidad	✓
	J4/42		
	J4/42	No	✓
	J1/B	continuidad	
	J4/43	No	✓
	J1/B	continuidad	

SUBRACK 1 P/N: A4 S/N: 01			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
<b>PASO 12:</b> - Conmutar a la posición NVG el interruptor SW08. - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/42 J1/B	Continuidad	✓
	J4/43 J1/B	No continuidad	✓
<b>PASO 13:</b> - Conmutar a la posición TEST el interruptor SW08. - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/42 J1/B	No continuidad	✓
	J4/43 J1/B	Continuidad	✓
	J4/44 J1/A	No continuidad	✓
	J4/45 J1/B	No continuidad	✓
<b>PASO 14:</b> - Conmutar a la posición CLEAR el interruptor SW05. - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/44 J1/A	Continuidad	✓
	J4/45 J1/B	Continuidad	✓
<b>PASO 15:</b> - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/46 J4/52	No continuidad	✓
	J4/47 J4/52	No continuidad	✓
	J4/48 J4/52	No continuidad	✓
	J4/49 J4/52	No continuidad	✓

SUBRACK 1 P/N: A4 S/N: 01			
PROCEDIMIENTO DE PRUEBA	CONECTOR/PIN	VALOR ESPERADO	PASA - FALLA
	J4/50	No continuidad	✓
	J4/52		
	J4/51	No continuidad	✓
	J4/52		
<b>PASO 16:</b> - Colocar las minihembrillas PS60-PS32, PS59-PS31, PS58-PS30, PS57-PS29, PS56-PS28 y PS55-PS27. - Aplicar las sondas del multímetro a las parejas de pines indicadas en cada caso.	J4/46	Continuidad	✓
	J4/52		
	J4/47	Continuidad	✓
	J4/52		
	J4/48	Continuidad	✓
	J4/52		
	J4/49	Continuidad	✓
	J4/52		
	J4/50	Continuidad	✓
	J4/52		
	J4/51	Continuidad	✓
	J4/52		

<b>SUBRACK 1</b>		<b>P/N: A4 S/N: 01</b>	
<b>PROCEDIMIENTO DE PRUEBA</b>	<b>CONECTOR/PIN</b>	<b>VALOR ESPERADO</b>	<b>PASA - FALLA</b>
<p><b>PASO 17:</b></p> <ul style="list-style-type: none"> <li>- Regular la primera salida de la fuente de tensión de salida dual 0-30VDC regulable para suministrar una tensión de +28VDC.</li> <li>- Aplicar el polo positivo (+) de la primera salida de la fuente de tensión dual 0-30VDC regulable al pin A del conector J1.</li> <li>- Aplicar el polo negativo (-) de la primera salida de la fuente de tensión dual 0-30VDC regulable al pin B del conector J1.</li> </ul>	<p>LEDG1</p> <p>LEDG2</p> <p>LEDG3</p> <p>LEDG4</p> <p>LEDG5</p> <p>LEDG6</p> <p>LEDG7</p> <p>LEDG8</p> <p>LEDG9</p> <p>LEDG10</p>	<p>Apagados</p>	<p>✓</p>

<b>SUBRACK 1</b>		<b>P/N: A4 S/N: 01</b>	
<b>PROCEDIMIENTO DE PRUEBA</b>	<b>CONECTOR/PIN</b>	<b>VALOR ESPERADO</b>	<b>PASA - FALLA</b>
<p><b>PASO 18:</b></p> <ul style="list-style-type: none"> <li>- Colocar los puentes de las minihembrillas PS07-PS08 y PS15-PS16.</li> <li>- Aplicar el polo negativo (-) de la primera salida de la fuente de tensión de salida dual 0-30VDC regulable a los siguientes pines: <ul style="list-style-type: none"> <li>· Conector J2, pin 1</li> <li>· Conector J2, pin 3</li> <li>· Conector J4, pin 5</li> <li>· Conector J4, pin 7</li> <li>· Conector J2, pin 9</li> <li>· Conector J2, pin 11</li> <li>· Conector J2, pin 13</li> <li>· Conector J2, pin 15</li> <li>· Conector J4, pin 24</li> <li>· Conector J4, pin 26</li> </ul> </li> </ul>	LEDG1 LEDG2 LEDG3 LEDG4 LEDG5 LEDG6 LEDG7 LEDG8 LEDG9 LEDG10	Encendidos	✓
<p><b>PASO 19:</b></p> <ul style="list-style-type: none"> <li>- Desconectar el polo negativo (-) de la primera salida de la fuente de tensión de salida dual 0-30VDC regulable de todos los pines <b>salvo del pin B del conector J1</b>.</li> <li>- Regular la segunda salida de la fuente de tensión de salida dual 0-30VDC regulable para suministrar</li> </ul>	LEDG1 LEDG2 LEDG3 LEDG4 LEDG5 LEDG6 LEDG7 LEDG8 LEDG9	Apagados	✓

<b>SUBRACK 1</b>		<b>P/N: A4 S/N: 01</b>	
<b>PROCEDIMIENTO DE PRUEBA</b>	<b>CONECTOR/PIN</b>	<b>VALOR ESPERADO</b>	<b>PASA - FALLA</b>
<p>una tensión de +5VDC.</p> <p>- Conectar el polo negativo (-) de la segunda salida de la fuente de tensión de salida dual 0-30VDC regulable con el polo negativo (-) de la primera salida de la fuente de tensión de salida dual 0-30VDC regulable, manteniendo la conexión de esta última con el pin B del conector J1.</p> <p>- Aplicar el polo positivo (+) de la segunda salida de la fuente de tensión de salida dual 0-30VDC regulable a los siguientes pines:</p> <ul style="list-style-type: none"> <li>· Conector J2, pin 1</li> <li>· Conector J2, pin 3</li> <li>· Conector J4, pin 5</li> <li>· Conector J4, pin 7</li> <li>· Conector J2, pin 9</li> <li>· Conector J2, pin 11</li> <li>· Conector J2, pin 13</li> <li>· Conector J2, pin 15</li> <li>· Conector J4, pin 24</li> <li>· Conector J4, pin 26</li> </ul>	LEDG10		

Tabla 27 Resultados funcionales "Subrack 1"



## **Anexo VIII. Planos de los bancos de integración.**

### **AVIII.i. Introducción**

La descripción física de los bancos es realizada por la empresa subcontratada, que será la que especifique las particularidades y descripción completa de equipos empleados dentro del banco, así como los planos. La representación de planos se realizará tanto en *layout* frontal y trasero, como completo.

### **AVIII.ii. Descripción del banco**

El Banco de integración del Sistema de Comunicaciones estará formado por armarios comerciales de 19” de ancho, 42U de altura y 900mm de fondo. Dispondrán de cáncamos en su parte superior con el fin de facilitar su manejo mediante elementos elevadores, sistema de evacuación de calor en su parte superior formado por un conjunto de ventiladores controlados por un termostato ajustable, ruedas con sistema de frenada, acceso por su base que evitando la entrada de polvo permita el paso de los cableados desde/hacia el falso suelo y puerta trasera dejando un acceso en su parte inferior para paso de cables.

Se dispondrá de una mesa de trabajo solidaria a los armarios con una profundidad máxima de 500mm y una longitud de al menos la anchura de tres armarios (uno de ellos deberá ser el denominado *Power Supply & SEAS*) . Será suficientemente rígida y se apoyará en el suelo en al menos dos puntos que no interferirán en el acceso a los *subracks*. Dada la necesidad podrá ser desmontada con facilidad.

Dispone de múltiples *subracks*, dependiendo de la cantidad de equipos que se necesiten en el sistema. Una posible disposición de *subracks* se presenta en las siguientes figuras:

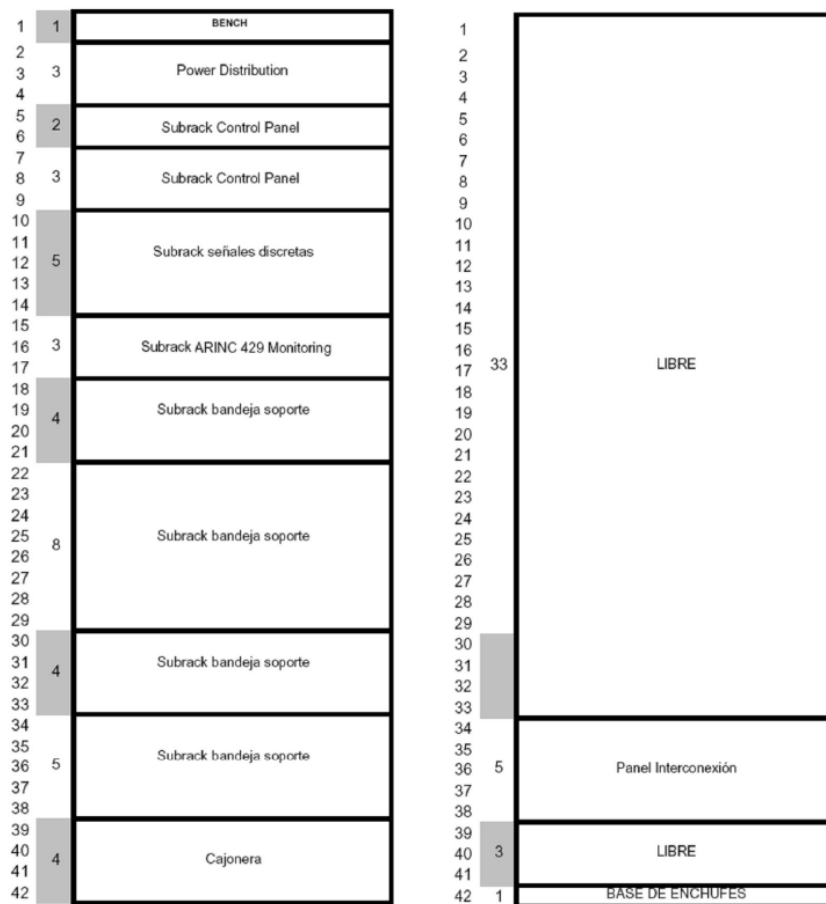
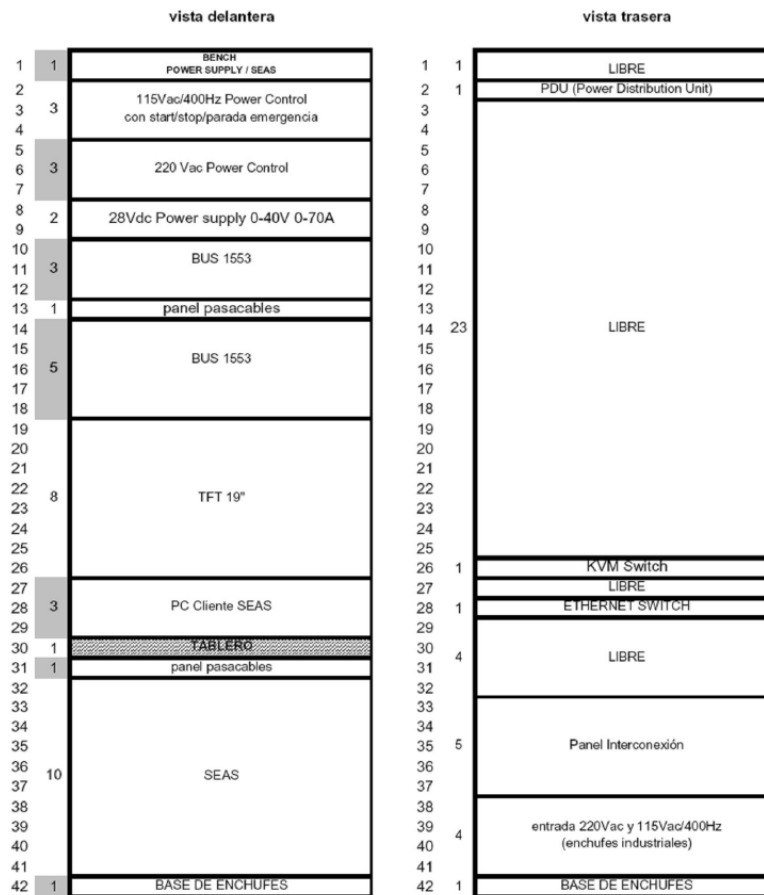


Figura 111 Subrack 1



**Figura 112 Subrack 2**

## **Anexo IX. Teoría sobre el calentamiento global**

### **AIX.i. Introducción**

A lo largo de la historia, el clima siempre ha variado, pero en el último siglo, este proceso de cambio se ha incrementado de manera no prevista. Muchos científicos coinciden en que el calentamiento global está ligado directamente al aumento de emisiones de CO<sub>2</sub> [54] desde la revolución industrial.

Este calentamiento global o cambio climático preocupa debido a los posibles problemas provocados en la naturaleza debido a una variación de la naturaleza no provocada directamente por los ciclos terrestres, si no por la mano del hombre.

### **AIX.ii. Fenómeno Natural**

El efecto invernadero es un efecto natural provocado por diferentes gases existentes en la atmósfera, y que provocan el aumento de la temperatura en el planeta con respecto a la que existiría únicamente por la acción directa del Sol. Uno de los gases fundamentales de este efecto, es el CO<sub>2</sub>.

La Tierra no es el único planeta en el que se registra el efecto invernadero. El planeta por excelencia donde este efecto se hace extremadamente notable sería Venus. La temperatura de Venus debida a la acción únicamente del Sol, debería ser en torno a los 70° C, sin embargo, debido a una concentración del 97% de CO<sub>2</sub> en su atmósfera, la temperatura en la superficie llega a los 450° C.

En la Tierra, gracias al efecto invernadero, la temperatura media del planeta pasa de -18° C a 15° C, lo que permite unas condiciones mucho más favorables para la vida.

Sin embargo, se sabe que la Tierra no ha permanecido siempre estable, si no que durante unos 150 mil años, se mantuvo en la conocida como época glacial. En esta época, la temperatura media del planeta era de 5° C, haciendo más difícil, pero posible, la vida. Estos ciclos naturales están muy estudiados y se sabe que responden habitualmente a los mismos tiempos y períodos de cambio, por eso, lo que ahora preocupa, es una variación anómala en estos ciclos naturales.

### **AIX.iii. Organización Internacional**

La comunidad internacional se ha organizado desde 1990 en diferentes organizaciones. En 1988 se creó el Panel Intergubernamental sobre el Cambio Climático (IPCC). El objetivo del IPCC es evaluar el riesgo del cambio climático originado por las actividades humanas, y sus informes se basan en publicaciones de revistas técnicas y científicas contrastadas. Cuenta con el trabajo de más de dos mil científicos provenientes de cien países.

El IPCC se ha convertido en la organización fundamental de información sobre cambio climático, compartiendo en 2008 el Nobel de la Paz junto con Al Gore. Sin embargo, el pacto que vincula legalmente a los países, y que más relevancia a tenido en estos últimos años, es el **Protocolo de Kyoto**.

#### **AIX.iii.a. Protocolo de Kyoto**

Este instrumento se encuentra dentro del marco de la Convención Marco de las Naciones Unidas sobre el Cambio Climático (CMNUCC), suscrita en 1992 dentro de lo que se conoció como la Cumbre de la Tierra de Río de Janeiro. El protocolo vino a dar fuerza vinculante a lo que en ese entonces no pudo hacer la CMNUCC.

El **Protocolo de Kyoto** sobre el cambio climático es un pacto internacional que tiene por objetivo reducir las emisiones de seis gases provocadores del Calentamiento Global: dióxido de carbono (CO<sub>2</sub>), gas metano (CH<sub>4</sub>) y óxido nitroso (N<sub>2</sub>O), además de tres gases industriales fluorados:

Hidrofluorocarbonos (HFC), Perfluorocarbonos (PFC) y Hexafluoruro de azufre ( $\text{SF}_6$ ), en un porcentaje aproximado de un 5%, dentro del periodo que va desde el año 2008 al 2012, en comparación a las emisiones al año 1990. Un total de 163 países ractificaron el pacto, aunque dentro de este pacto no entraron países en vías de desarrollo como China o India, que en un futuro, probablemente serán tanto o más importantes que otros.

Cada país no debe reducir sus emisiones de gases regulados en un 5%, sino que este es un porcentaje global y, por el contrario, cada país obligado por Kyoto tiene sus propios porcentajes de emisión que debe disminuir. Se tomaron como referencia de emisiones, las de 1990, por lo que se medirá entre 2008 y 2012 las emisiones medias, para referenciarlas a 1990 y comprobar el cumplimiento del pacto.

La Unión Europea tiene fijada una reducción del 8%, si bien se realizó un reparto entre sus países miembros, de forma, que por ejemplo a España, se le consentiría un aumento en sus emisiones de 15% partiendo como base de sus emisiones en 1990. El problema para España radica, en que, hasta la fecha, estas emisiones han aumentado en un 53%, lo que complica en gran medida el cumplimiento del **Protocolo de Kyoto**.

Estados Unidos es otro de los grandes protagonistas en el **Protocolo de Kyoto**, ya que aunque firmo el acuerdo en 1998, lo rechazó posteriormente, y hasta el momento se niega a ratificarlo. Los miembros del tratado están estudiando nuevas fórmulas para que Estados Unidos y otros países muy contaminantes en vías de desarrollo, firmen el acuerdo y reduzcan sus emisiones.

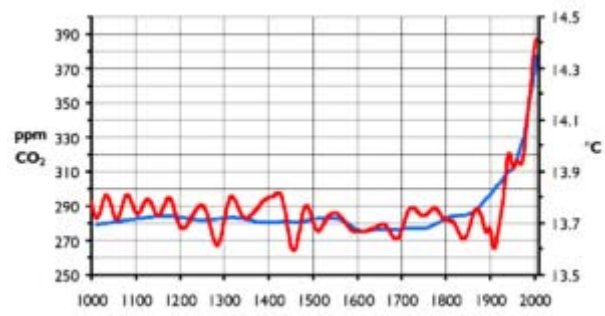


Figura 113 Relación de acumulación de CO2 y aumento de temperatura [54]

## Anexo X. Código de interacción con BlackBox

### AX.i. Introducción

Para la interacción con la *BlackBox* es necesario un programa que evite la necesidad de uso de un explorador *web* siempre que se quiera modificar la configuración. El código y la obtención de éste se muestra a continuación.

### AX.ii. Código

La forma en que se gestiona las capacidades de la *BlackBox* es a través de una página *web* que nos sirve el propio equipo. Por lo tanto, se puede utilizar el protocolo *http* para realizar las peticiones necesarias.

Los comandos no son muy diversos, apagado y encendido de las diferentes unidades de salida del *BlackBox*, por lo tanto habrá que conocer cuál es esta petición.

Utilizando *Ethereal* para ver los comandos enviados mediante el *WEB server* se obtiene el siguiente resultado:

```
<?xml version="1.0" ?>
<pdml version="0" creator="ethereal/0.10.9">
  <packet>
    <proto name="http" showname="Hypertext Transfer Protocol" size="505" pos="54">
      <field show="GET /rpmstatus.cgi?all=on&iswid=0 HTTP/1.1\r\n" size="44" pos="54"
        value="474554202f72706d7374617475732e6367693f616c6c3d6f6e2669737769643d302
        0485454502f312e310d0a">
        <field name="http.request.method" showname="Request Method: GET" size="3"
          pos="54" show="GET" value="474554" />
      </field>
      <field name="http.host" showname="Host: xxxxsw01_rpc\r\n" size="20" pos="98"
        show="xxxsw01_rpc" value="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx" />
      <field name="http.user_agent" showname="User-Agent: Mozilla/5.0 (Windows; U;
        Windows NT 5.0; es-ES; rv:1.9.0.5) Gecko/2008120122 Firefox/3.0.5\r\n" size="104"
        pos="118" show="Mozilla/5.0 (Windows; U; Windows NT 5.0; es-ES; rv:1.9.0.5)
        Gecko/2008120122 Firefox/3.0.5"
        value="557365722d4167656e743a204d6f7a696c6c612f352e30202857696e646f77733b2
```



```

0553b2057696e646f7773204e5420352e303b2065732d45533b2072763a312e392e302e3
529204765636b6f2f323030383132303132322046697265666f782f332e302e350d0a" />
    <field          name="http.accept"          showname="Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n"          size="73"
pos="222"      show="text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8"
value="4163636570743a20746578742f68746d6c2c6170706c69636174696f6e2f7868746
d6c2b786d6c2c6170706c69636174696f6e2f786d6c3b713d302e392c2a2f2a3b713d302
e380d0a" />
<field name="http.accept_language" showname="Accept-Language: es-es,es;q=0.8,en-
us;q=0.5,en;q=0.3\r\n"          size="54"          pos="295"          show="es-es,es;q=0.8,en-
us;q=0.5,en;q=0.3"
value="4163636570742d4c616e67756167653a2065732d65733b713d302e382c65
6e2d75733b713d302e352c656e3b713d302e330d0a" />
<field name="http.accept_encoding" showname="Accept-Encoding: gzip,deflate\r\n"
size="31"          pos="349"          show="gzip,deflate"
value="4163636570742d456e636f64696e673a20677a69702c6465666c6174650d0a" />
<field show="Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n" size="48" pos="380"
value="4163636570742d436861727365743a2049534f2d383835392d312c7574662d383b
713d302e372c2a3b713d302e370d0a" />
    <field          show="Keep-Alive: 300\r\n"          size="17"          pos="428"
value="4b6565702d416c6976653a203330300d0a" />
<field name="http.connection" showname="Connection: keep-alive\r\n" size="24"
pos="445"          show="keep-alive"
value="436f6e6e656374696f6e3a206b6565702d616c6976650d0a" />
<field name="http.referer" showname="Referer: http://xxxxsw01_rpc/rpm_status.html\r\n"
size="45"          pos="469"          show="http://xxxxsw01_rpc/rpm_status.htm"
value="xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx" />
<field name="http.request" showname="Request: True" hide="yes" size="0" pos="54"
show="1" />
</proto>
</packet>
</pdml>

```

Por lo tanto la petición que se debe realizar es:

*GET /rpmstatus.cgi?all=on|off&iswid=n HTTP/1.1*

- **on|off:** define el *status* encendido/apagado de la salida concreta.
- **n:** será la salida elegida sobre la que actuar.

## Blackbox\_controller.cpp

```
#include "stdafx.h"
#include "stdio.h"

#include <winsock.h>
#pragma comment(lib, "ws2_32.lib")

int main(int argc, char* argv[])
{
    WSADATA wsaData;
    SOCKET SendSocket;
    sockaddr_in RecvAddr;
    struct hostent *he;
    int Puerto = 80;
    int bytes_sent;

    char *host_name = argv[1];
    char comando[200];

    comando[0] = '\0';

    strcat(comando, "GET /rpmstatus.cgi?all=");
    strcat(comando, argv[2]);
    strcat(comando, "&iswid=");
    strcat(comando, argv[3]);
    strcat(comando, " HTTP/1.1\r\n\r\n");

    WSStartup(MAKEWORD(2,2), &wsaData);
    SendSocket = socket( AF_INET, SOCK_STREAM, 0);

    RecvAddr.sin_family = AF_INET;
    RecvAddr.sin_port = htons(Puerto);

    if ((he=gethostbyname(host_name)) == NULL) {
        printf("error en gethostbyname");
        exit(1);
    }

    RecvAddr.sin_addr = *((struct in_addr *)he->h_addr);

    if ( connect( SendSocket, (SOCKADDR*) &RecvAddr, sizeof(RecvAddr) )
    == SOCKET_ERROR){

        printf ("Error en connect");

    }

    bytes_sent = send(SendSocket, comando, strlen(comando), 0);

    Sleep (500);

    WSACleanup();
    return 0;
}
```

```
}
```

Blackbox\_controller.cpp

## Anexo XI. Pruebas de comprobación del banco

### AXI.i. Electrical Testing

El test eléctrico del banco es responsabilidad de la empresa subcontratada. Debe ser realizado por la subcontrata y personal de EADS podrá supervisar el procedimiento siempre que quiera.

Las pruebas eléctricas que realizaremos sobre el banco se definen a continuación:

#### AXI.i.a. Prueba de continuidad

PRUEBA DE CONTINUIDAD	
<u>Tipo de Prueba:</u>	Continuidad.
<u>Condiciones de la Prueba:</u>	Condiciones normales de laboratorio.
<u>Equipo de Prueba:</u>	Multímetro de propósito general en función medición de resistencias.
<u>Propósito de la Prueba:</u>	<p>El propósito de esta prueba es:</p> <p>Comprobar las conexiones dentro de cada uno de los mazos que conforman el conjunto de mazos.</p> <p>Comprobar que no existen falsos contactos ni conexiones abiertas.</p> <p>Será necesario efectuar un chequeo pin a pin de <b>todos los conectores</b> de todas las unidades como única forma de descartar conexiones no deseables que no puedan ser detectadas mediante la comprobación de acuerdo a su documentación de diseño.</p>

PRUEBA DE CONTINUIDAD				
<u>Realización de la Prueba:</u>	<p><b>Paso 1:</b> Encender el instrumento en modo medición de resistencias con escala de ohmios.</p> <p><b>Paso 2:</b> Unir los extremos de los conectores de prueba del instrumento para comprobar correcto funcionamiento y resistencia de las puntas de prueba.</p> <p><b>Paso 3:</b> Conectar los cabezales de medida al circuito que se desea comprobar.</p> <p><b>Paso 4:</b> Comprobar valor de resistencia mostrado por el display del instrumento.</p> <p><b>Paso 5:</b> Retirar las conexiones de prueba.</p> <p><b>Paso 6:</b> Repetir los pasos del 3 al 5 para cada uno de los pines entre los que se desea realizar la prueba de continuidad.</p>			
<u>Objetivo de la Prueba:</u>	Comprobar que todas las conexiones descritas tienen una resistencia menor de 2 ohmios ( $\Omega$ ).			
<u>Criterio de Aceptación de la Prueba:</u>	<b>PASA</b>  $< 2 \Omega$		<b>NO PASA</b>  $\geq 2 \Omega$	
Equipo de Prueba	Tipo	Modelo y P/N	Fabricante	Fecha Cal.
Instrumento 1	Multímetro	FLUKE 189 ELAD-M0025	FLUKE	24/05/2008

Tabla 28 Prueba de continuidad

**AXI.i.b. Prueba de continuidad cruzada**

<b>PRUEBA DE CONTINUIDAD CRUZADA</b>	
<u>Tipo de Prueba:</u>	Continuidad cruzada.
<u>Condiciones de la Prueba:</u>	Condiciones normales de laboratorio.
<u>Equipo de Prueba:</u>	Multímetro de propósito general en función medición de resistencias.
<u>Propósito de la Prueba:</u>	<p>El propósito de esta prueba es comprobar el aislamiento del pin a prueba respecto al resto de pines del mazo, a excepción del pin especificado en el documento de diseño.</p> <p>Será necesario efectuar un chequeo pin a pin de <b>todos los conectores</b> de todas las unidades como única forma de descartar conexiones no deseables que no puedan ser detectadas mediante la comprobación de acuerdo a su documentación de diseño.</p>
<u>Realización de la Prueba:</u>	<p><b>Paso 1:</b> Encender el instrumento en modo medición de resistencias con escala de Megaohmios.</p> <p><b>Paso 2:</b> Unir los extremos de los conectores de prueba del instrumento para comprobar correcto funcionamiento y resistencia de las puntas de prueba.</p> <p><b>Paso 3:</b> Conectar los cabezales de medida al circuito que se desea comprobar.</p> <p><b>Paso 4:</b> Comprobar valor de resistencia mostrado por el display del instrumento.</p>

PRUEBA DE CONTINUIDAD CRUZADA				
	<p><b>Paso 5:</b> Retirar las conexiones de prueba.</p> <p><b>Paso 6:</b> Repetir los pasos del 3 al 5 para cada uno de los pines entre los que se desea realizar la prueba de continuidad cruzada.</p>			
<u>Objetivo de la Prueba:</u>	Comprobar que todas las conexiones descritas tienen una resistencia mayor que 20 Megaohmios (MΩ).			
<u>Criterio de Aceptación de la Prueba:</u>	<p><b>PASA</b></p> <p>&gt; 20 MΩ</p>		<p><b>NO PASA</b></p> <p>≤ 20 MΩ</p>	
Equipo de Prueba	Tipo	Modelo y P/N	Fabricante	Fecha Cal.
Instrumento 1	Multímetro	FLUKE 189 ELAD-M0025	FLUKE	24/05/2008

Tabla 29 Prueba de continuidad cruzada

**AXI.i.c. Prueba de aislamiento respecto a carcasa**

PRUEBA DE AISLAMIENTO RESPECTO A CARCASA	
<u>Tipo de Prueba:</u>	Aislamiento respecto a carcasa.
<u>Condiciones de la Prueba:</u>	Condiciones normales de laboratorio.
<u>Equipo de Prueba:</u>	Multímetro de propósito general en función medición de resistencias.
<u>Propósito de la Prueba:</u>	El propósito de esta prueba es comprobar que se cumplen las

PRUEBA DE AISLAMIENTO RESPECTO A CARCASA		
	<p>normas seguridad para equipos y personas, verificando que no existen conexiones erróneas ni derivaciones de corriente que den lugar a cortocircuitos y fallos antes de energizar la unidad.</p>	
<u>Realización de la Prueba:</u>	<p><b>Paso 1:</b> Encender el instrumento en modo medición de resistencias con escala de Megaohmios.</p> <p><b>Paso 2:</b> Unir los extremos de los conectores de prueba del instrumento para comprobar correcto funcionamiento y resistencia de las puntas de prueba.</p> <p><b>Paso 3:</b> Conectar los cabezales de medida al circuito que se desea comprobar.</p> <p><b>Paso 4:</b> Comprobar valor de resistencia mostrado por el display del instrumento.</p> <p><b>Paso 5:</b> Retirar las conexiones de prueba.</p> <p><b>Paso 6:</b> Repetir los pasos del 3 al 5 para cada uno de los pines entre los que se desea realizar la prueba de aislamiento respecto a carcasa.</p>	
<u>Objetivo de la Prueba:</u>	<p><b>Comprobar que todas las conexiones descritas tienen una resistencia mayor que 20 Megaohmios (MΩ).</b></p>	
<u>Criterio de Aceptación de la Prueba:</u>	<p><b>PASA</b></p> <p>&gt; 20 MΩ</p>	<p><b>NO PASA</b></p> <p>≤ 20 MΩ</p>



PRUEBA DE AISLAMIENTO RESPECTO A CARCASA				
Equipo de Prueba	Tipo	Modelo y P/N	Fabricante	Fecha Cal.
Instrumento 1	Multímetro	FLUKE 189 ELAD-M0025	FLUKE	24/05/2008

Tabla 30 Prueba de aislamiento respecto a carcasa

#### AXI.ii. Integración de SEAS

La integración de SEAS en banco es responsabilidad de la empresa subcontratada. Podrá ser supervisado por personal de EADS. Corresponde con las pruebas funcionales del banco.

#### AXI.ii.a. Prueba de inspección visual

PRUEBA DE INSPECCIÓN VISUAL	
<u>Tipo de Prueba:</u>	Inspección visual.
<u>Condiciones de la Prueba:</u>	Condiciones normales de laboratorio. Unidad bajo prueba completamente desenergizada y aislada. Tapa superior desmontada.
<u>Equipos de Prueba:</u>	N/A
<u>Propósito de la Prueba:</u>	<p>El propósito de esta prueba es:</p> <ul style="list-style-type: none"> <li>Comprobar el correcto acabado del subrack, así como que éste no presenta aristas cortantes o rebabas que puedan producir daños físicos a las personas que operen con él en su entorno o mantenimiento.</li> <li>Comprobar el correcto montaje mecánico del subrack, correspondencia con esquemas, así como el grapado o soldado de los componentes que la componen.</li> </ul>

PRUEBA DE INSPECCIÓN VISUAL	
<u>Preparación de la Prueba:</u>	N/A
<u>Objetivo de la Prueba:</u>	<p><b>Comprobar el correcto acabado superficial del subrack.</b></p> <ul style="list-style-type: none"> <li>- Comprobar la correcta serigrafía de los componentes situados en la parte frontal y trasera del subrack.</li> <li>- Comprobar que todos los componentes están integrados correctamente dentro del subrack.</li> <li>- Comprobar el correcto montaje mecánico del subrack.</li> <li>- Comprobar el correcto grapado y/o soldado de los componentes.</li> <li>- Comprobar la correspondencia de tamaño entre galgas y cables.</li> <li>- Comprobar la correspondencia entre conectores y el subrack.</li> <li>- Comprobar el correcto etiquetado del subrack.</li> <li>- Comprobar correcto dimensionamiento del subrack.</li> </ul>
<u>Criterio de Aceptación de la Prueba:</u>	Definido en cada caso particular.

Tabla 31 Prueba de inspección visual

#### AXI.ii.b. Prueba funcional

PRUEBA FUNCIONAL	
<u>Tipo de Prueba:</u>	Funcional
<u>Condiciones de la Prueba:</u>	Condiciones normales de laboratorio. Unidad bajo prueba completamente desenergizada y aislada.
<u>Equipos de Prueba:</u>	Fuente de tensión de salida dual 0-30 VDC regulable y multímetro digital.
<u>Propósito de la</u>	Comprobar que la unidad cumple los requisitos funcionales descritos en su especificación o esquema funcional.

PRUEBA FUNCIONAL				
<u>Prueba:</u>				
<u>Preparación de la Prueba:</u>	<p><b>Paso 1:</b> Retirar todos los puentes de minihembrillas.</p> <p><b>Paso 2:</b> Conmutar todos los interruptores a la posición OFF, NORMAL o MANUAL según corresponda.</p>			
<u>Objetivo de la Prueba:</u>	Comprobar la correcta funcionalidad de todos y cada uno de los elementos de la unidad.			
<u>Planos de Referencia:</u>	A4-2A076-00H02, A4-2A076-00H03, A4-2A076-00H04 y A4-2A076-00H05.			
<u>Criterio de Aceptación de la Prueba:</u>	Definido en cada caso particular.			
Equipo de Prueba	Tipo	Modelo y P/N	Fabricante	Fecha Cal.
INSTRUMENTO 1	Multímetro	FLUKE 189 ELAD-M0025	FLUKE	24/05/2008
INSTRUMENTO 2	Fuente de tension de salida dual 0-30 VDC regulable	E3647A ELAD-OS03	AGILENT TECHNOLOGIES, INC.	22/07/2008

Tabla 32 Prueba funcional

## Anexo XII. Código de integración de SVN

### AXII.i. Update

```
svn_error_t *
svn_ra_actualiza(char *username,
                char *password,
                const char *config_dir,
                const char *path,
                const char *url)
{
    svn_ra_session_t *session;
    svn_wc_adm_access_t *adm_access;
    svn_client_ctx_t *ctx;
    svn_revnum_t head;
    apr_hash_t *config;
    svn_error_t *err;
    apr_hash_t *props;
    svn_revnum_t base_revision;
    apr_pool_t *pool = apr_create_pool(NULL);
    apr_array_header_t **result_revs;
    svn_opt_revision_t peg_revision;

    SVN_ERR(svn_fs_initialize(pool));
    SVN_ERR(svn_client_create_context(&ctx, pool));

    /* Obtenemos la autorización de sesión para realizar el checkout */
    svn_ra_callbacks_t *callbacks = apr_palloc(pool, sizeof(*callbacks));
    svn_cmdline_setup_auth_baton(&callbacks->auth_baton, TRUE,
                                username, password,
                                NULL, FALSE, NULL, NULL, NULL, pool);

    /* Comprobamos si tenemos acceso a la Working Copy */
    err = svn_wc_adm_probe_open3(&adm_access, NULL, url,
                                FALSE, adm_lock_level,
                                ctx->cancel_func, ctx->cancel_baton,
                                pool);

    if(err){
        /* Abrimos la sesión y realizamos Checkout */
    }
}
```

```

SVN_ERR(svn_ra_open(&session, url,
callbacks,NULL, config, pool));
SVN_ERR(svn_ra_get_latest_revnum(session, &head, pool));
peg_revision.kind = svn_opt_revision_unspecified;
SVN_ERR(svn_client_checkout3(&head,url,path,&peg_revision,NULL,
                                svn_depth_infinity,TRUE,TRUE,
                                ctx,pool);

}else{
    /* Obtenemos la configuración de la working copy y abrimos sesión */
    SVN_ERR(svn_config_get_config(&config, config_dir, pool));
    SVN_ERR(svn_ra_open(&session, url,
callbacks,NULL, config, pool));
    SVN_ERR(svn_ra_get_latest_revnum(session, &head, pool));
    SVN_ERR(svn_wc_prop_list(&props,path,&adm_access,pool);
    base_revision = apr_hash_get(props, SVN_PROP_BASE_REVISION,
                                APR_HASH_KEY_STRING);

    if(base_revision<head){
        /* Si no tenemos la ultima versión, actualizamos */
        SVN_ERR(svn_client_update3(result_revs,path,
                                &peg_revision,svn_depth_infinity,FALSE,
                                TRUE,TRUE,ctx,pool);
    }
}
}
}

```

El código sirve tanto para crear la WC del cliente SEAS, como para realizar la WC del servidor SEAS en el servidor de ficheros, ya que lo que pasamos como parámetro es un *path* que no tiene porque estar en nuestro disco local.

## AXII.ii. Commit

```

svn_error_t *
svn_ra_commit(char *username,
              char *password,
              const char *config_dir,
              const char *path,
              const char *url)
{

```

```

svn_ra_session_t *session;
svn_client_ctx_t *ctx;
svn_revnum_t head;
apr_hash_t *config;
svn_revnum_t base_revision;
svn_commit_info_t *commit_info = NULL;

apr_pool_t *pool = apr_create_pool(NULL);
SVN_ERR(svn_fs_initialize(pool));
SVN_ERR(svn_client_create_context(&ctx, pool));

/* Obtenemos la autorización de sesión para realizar el commit */
svn_ra_callbacks_t *callbacks = apr_palloc(pool, sizeof(*callbacks));
svn_cmdline_setup_auth_baton(&callbacks->auth_baton, TRUE,
                             username, password,
                             NULL, FALSE, NULL, NULL, NULL, pool);
SVN_ERR(svn_config_get_config(&config, config_dir, pool));
SVN_ERR(svn_ra_open(&session, url, callbacks, NULL, config, pool));
SVN_ERR(svn_ra_get_latest_revnum(session, &head, pool));
/* Realizamos commit */
SVN_ERR(svn_client_commit(&commit_info, path, FALSE, ctx, pool));
}

```

A partir de los códigos anteriores, y realizando las pruebas y modificaciones necesarias, se construye la librería dinámica a través de la que enlazar el subsistema de ensayo con el repositorio de SVN.

## Anexo XIII. Script de apagado remoto

### AXIII.i. Banco con SAI y RPC

```
@echo off

:: Hay dos opciones, o BLACKBOX, o PULIZZI
:: --- set REMOTE_SHUTDOWN_MODE=BLACKBOX --- ::
set REMOTE_SHUTDOWN_MODE=%1

:: --- set REMOTE_SHUTDOWN_HOSTNAME=a400_fire_rpc --- ::
set REMOTE_SHUTDOWN_HOSTNAME=%2

::-----::
::----- Apagado de banco -----::
::-----::

if "%REMOTE_SHUTDOWN_MODE%" == "NONE" goto end

if          "%REMOTE_SHUTDOWN_MODE%"          ==          "BLACKBOX"          blackbox_controller
%REMOTE_SHUTDOWN_HOSTNAME% off 1

if "%REMOTE_SHUTDOWN_MODE%" == "PULIZZI" pulizzi %REMOTE_SHUTDOWN_HOSTNAME% -1
```

### AXIII.ii. Banco con RPC

```
@echo off

set SERVER_NAME=%1

:: Hay dos opciones, o BLACKBOX, o PULIZZI
:: --- set REMOTE_SHUTDOWN_MODE=BLACKBOX --- ::
set REMOTE_SHUTDOWN_MODE=%2

:: --- set REMOTE_SHUTDOWN_HOSTNAME=a400_fire_rpc --- ::
set REMOTE_SHUTDOWN_HOSTNAME=%3

::-----::
::----- Recoger informacion de los procesadores -----::
::-----::

:: - Se crea el fichero de procesadores.txt con los procesadores especificados
:: en el HW Config del server

if not exist \\%SERVER_NAME%\seas3_srv$\CONFIG\HW_CONFIG_SEAS.XML goto not_ready

info_processors %SERVER_NAME% > procesadores.txt 2> nul
```

```

::-----::
::--- Apagar todos los procesadores del banco -----::
::-----::

FOR /F "tokens=1" %%i IN (procesadores.txt) DO (

psshutdown -c -f -t 10 \\%%i > nul 2> nul

)

::-----::
::----- Comprobación procesadores -----::
::-----::

echo [%DATE% - %TIME%] Comprobando estado de procesadores
echo.

:alguno_encendido

set ALGUNO_ENCENDIDO=0

FOR /F "tokens=1" %%i IN (procesadores.txt) DO (

ping -n 1 %%i | find "TTL" > nul

if not errorlevel 1 (

echo [%DATE% - %TIME%] Procesador %%i encendido.
set ALGUNO_ENCENDIDO=1

)

)

sleep 5

if %ALGUNO_ENCENDIDO%==1 goto alguno_encendido

set ALGUNO_ENCENDIDO=

del procesadores.txt

::-----::
::----- Apagado de banco -----::
::-----::

if "%REMOTE_SHUTDOWN_MODE%" == "NONE" goto end

if          "%REMOTE_SHUTDOWN_MODE%"          ==          "BLACKBOX"          blackbox_controller
%REMOTE_SHUTDOWN_HOSTNAME% off 1

```



```

if "%REMOTE_SHUTDOWN_MODE%" == "PULIZZI" pulizzi %REMOTE_SHUTDOWN_HOSTNAME% -I

goto end

:end

```

Como se ve en el código, es necesario crear un fichero *procesadores.txt* a partir del fichero *HW\_CONFIG\_SEAS.XML*. Para ello se emplea la aplicación *info\_processors.exe*. Con esta aplicación se obtienen los nombres de servidor de los equipos SEAS que deben ser apagados correctamente.

### **AXIII.iii. Banco sin recursos adicionales**

```

@echo off

set SERVER_NAME=%I

::-----::
::----- Recoger informacion de los procesadores -----::
::-----::
:: - Se crea el fichero de procesadores.txt con los procesadores especificados
:: en el HW Config del server

if not exist \\%SERVER_NAME%\seas3_srv$\CONFIG\HW_CONFIG_SEAS.XML goto not_ready

info_processors %SERVER_NAME% > procesadores.txt 2> nul

::-----::
::---- Apagar todos los procesadores del banco -----::
::-----::

FOR /F "tokens=1" %%i IN (procesadores.txt) DO (

psshutdown -c -f -t 10 \\%%i > nul 2> nul

)

::-----::
::----- Comprobación procesadores -----::
::-----::

echo [%DATE% - %TIME%] Comprobando estado de procesadores
echo.

```

```
:alguno_encendido

set ALGUNO_ENCENDIDO=0

FOR /F "tokens=1" %%i IN (procesadores.txt) DO (

ping -n 1 %%i /find "TTL" > nul

if not errorlevel 1 (

echo [%DATE% - %TIME%] Procesador %%i encendido.
set ALGUNO_ENCENDIDO=1

)

)

sleep 5

if %ALGUNO_ENCENDIDO%==1 goto alguno_encendido

set ALGUNO_ENCENDIDO=0

del procesadores.txt

:end
```

## Capítulo 7. Bibliografía

- [1] [Conmutación de circuitos](#)
- [2] [GPIB Tutorial](#)
- [3] [Standard RS-232](#)
- [4] [“Puerto Serie RS-232”, José Manuel Murcia Barba, ITIG](#)
- [5] [Redes. La instalación física \(segunda parte\), A. Abad.](#)
- [6] Protocolo USB, Eric López Pérez, Ingeniería en Microcontroladores.
- [7] [Sistemas de instrumentación VXI.](#)
- [8] [Short Tutorial on VXI](#)
- [9] LXImate, “A practical guide to the LXI standard and Getting Started with LXI Devices”, Pickering Interfaces, Febrero 2008.
- [10] [Interfaz VME](#)
- [11] [“Control de versiones con Subversion, Revisión 3428”. Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato.](#)
- [12] [“Gestión de versiones con CVS y Subversion”, Fernando López Hernández para MacProgramadores, Klagenfurt Septiembre de 2007.](#)
- [13] [“Concurrent Versions System”, Copyright \(C\) 2005-2006 Derek Robert Price & Ximbiot and Copyright \(C\) 1998 Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111, USA](#)
- [14] [“Adding Version Control to Your Application with Subversion”, Garrett Rooney. CollabNet](#)
- [15] [Tigris.org, Open Source Software Engineering Tools. Subversion.](#)
- [16] [¿Qué es un CVS? Instalación de Subversion.](#)
- [17] A400M Aircraft System Design Route
- [18] “SEAS Overview”, Bartolomé Lozano Cerón, Noviembre 2007.
- [19] “SEAS Overview”, Bartolomé Lozano Cerón, Febrero 2003.
- [20] “SEAS3 Data Visualization User Guide”, Ángel Perea Velázquez, Enero 2004.
- [21] “SEAS3 Recording Definition User Guide”, Enrique Chicharro López, Octubre 2003.
- [22] “SEAS3 Sequencer User Guide”, Javier Piñeiro Rodríguez, Marzo 2006.
- [23] “SEAS3 Analysis User Guide”, Enrique Chicharro López, Octubre 2003.

- [24] "SEAS3 Simulations User Guide", Enrique Chicharro López, Diciembre 2005.
- [25] "Status of VME Technology", Hermann Strass. Sales Conference Europe, May 11-13, 2005.
- [26] ["What is PXI?", National Instruments.](#)
- [27] ["The PXI Modular Instrumentation Architecture"](#)
- [28] "ARINC Protocol Tutorial", Condor Engineering Inc.
- [29] "ARINC429 Commentary", SBS Technologies Inc.
- [30] "AFDX Bus Introduction, Overview", Military Transport Aircraft Division, EADS-CASA.
- [31] "AFDX Presentation", Vincent Buron, Airbus.
- [32] "Draft 3 of project paper 664, Aircraft Data Network, part 7. Avionics Full Duplex Switched Ethernet (AFDX) Network", Aeronautical Radio Inc., September 2004.
- [33] "MIL-STD-1553 Tutorial", Condor Engineering Inc.
- [34] "An Overview of MIL-STD-1553", AIM.
- [35] "Introduccion 1553", Military Transport Aircraft Division, EADS-CASA.
- [36] "Flying High with Stanag3910", Douglas Ullah and Hansjoerg Frey, AIM GmbH.
- [37] "Controller Area Network, CAN Bus", Military Transport Aircraft Division, EADS-CASA.
- [38] "Guide to digital interface standards for military avionic applications", Avionic Systems Standarisation Comitee.
- [39] [AIM Datasheet AMC1553-x](#)
- [40] [AIM Datasheet AMC429](#)
- [41] AIM Datasheet AVI3910
- [42] [NI PCI-CAN/XS2](#)
- [43] ["GPIB Tutorial"](#)
- [44] [Wikipedia: GPIB](#)
- [45] [Conector RS-232](#)
- [46] [PinOut RS-232](#)
- [47] ["SGI Origin 3900 Server User's Guide"](#)
- [48] ["Ethernet RJ-45 LoopBack, Crossed and Pinouts"](#)
- [49] [Conector USB tipo A](#)

- [50] ["Capitulo 3: Bus Serie Universal USB", Pablo Hoffman.](#)
- [51] ["MGE's Network Shutdown Solution"](#)
- [52] "Descripción de requerimientos HW para bancos de ensayo"
- [53] "Descripción de requerimientos SW para bancos de ensayo"
- [54] "The Environment Council Briefing Paper on Sustainable Events", Alison Bowman, Corporate Relations Coordinator, The Environment Council.
- [55] MGE UPS Datasheet
- [56] ["Reducción de los costos de pruebas con las tecnologías de conmutación de estado sólido", Nacional Instruments.](#)
- [57] ["Sensing and Control with VB Programming: Computer Sensing Non Mechanical Switching Devices part 2", CDLI.](#)
- [58] "Manual de desarrollo de bancos de integración", Enrique Martín, Bartolomé Lozano.